**WEB CRAWLER**

**A Thesis Submitted to the Department of information Technology, Jadavpur University in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Software Engineering**

Submitted by:

**Shirsangshu Dutta**

Examination Roll Number – M4SWE1620R

Registration Number – 79697 of 2001-2002

Under the Guidance of

**Mr Uttam Kumar Roy**

**H.O.D**

**Dept. of Information Technology**

**Jadavpur University**

**WEB CRAWLER**

**Contents**

**Acknowledgments**

**Abstract**

1. **Introduction 1**
   1. **What is web crawling…………………………………………………………………… 4**
   2. **Utility of web crawlers and scrapers………………………………………………………………..............................**[**5**](#Motivation)
   3. **indexing web pages........................................................**
2. **Social Media**
   1. **Social Network Analysis**
3. **Designing a web crawler**

**3.1 Design goals of a web crawler**

**3.2 Challenge in designing a crawler**

1. **Key factors for optimized crawler**
   1. **Page Selection**
   2. **Parallelization of Crawlers and distributed crawlers**
   3. **Freshness and Page refresh policy**
2. **Architecture of crawler**
   1. **Types of crawler**
   2. **Algorithm of crawler**
3. **Implementation challenges**
   1. **Solution to counter challenges**
4. **Test experiment**
   1. **Social media linkedin**
   2. **Survey of the technologies available for scraping**
   3. **Techniques and strategies attemped**
   4. **Observations from the experiment**
   5. **Metrics collected**
   6. **Analyzing data collected**
5. **Conclusion and further works**
6. **Citations and references**

**Acknowledgement**

I would like to express my deepest regards to Dr. Uttam Kumar Roy, Assistant Professor, Head of the Department of Information Technology, Jadavpur University for giving me the opportunity, freedom, support and above all guidance to work in the field of web crawling and giving me leads and assignments which provoke my thoughts and hence developed my interest in this subject .Also through the pursuit of this thesis the knowledge gained will further my career

I would like to thank my parents for providing all sorts of support throughout my career and studies

**Abstract**

Crawling the web has been an area of research and interest for long just because of the intrinsic value of the target data .The document retrieved in web search have been found to truthful in majority of cases .This has led to study for better understanding of the structure of web and the layout of documents and the information retrieval techniques so that more efficient algorithms can be developed .

The main application of web crawler is definitely search engines. The world wide web is massive and is growing at an unprecented 7.6 percent annually .It is only possible to keep with the growth and produce better search results only if its indexed properly .Web crawler or bots or spiders are used by large search engines for accurate information retrieval.They necessary do not cover the whole internet (for example google claims to index 35 trillion web pages which is on 4% of total internet) but the indexing techniques ensure that most relevant pages show up in search .The techniques for page selection,page revisit policy,estimation of freshness,crawling frequency are the areas of interest for studying crawling/scraping

The main objective of the paper is to study the various techniques that can used for mining the web for useful information.Here we focus on scraping the data i.e storing the data after information retrieval for some useful purpose .The same techniques that one uses in web crawling can be used for scraping data

The particular area of interest is social media as that is place where lot of people come together and voluntarily update their information .The main objective in social media platforms is to connect with other people having common areas of interests.So this provides a huge wealth of information worth mining . Just because of “network” thing in Social network it become an easy spot just to dive in and get the data as clusters of similar data is already there by virtue of its design

**Introduction**

**1.2 What is web crawler**

A web crawler, spider, or search engine bot downloads and indexes content from all over the Internet. The goal of such a bot is to learn what (almost) every webpage on the web is about, so that the information can be retrieved when it's needed

* 1. **Utility of web crawlers and scrapers**
* Support universal search engines (Google, Yahoo, MSN/Bing, etc.)
* Vertical (specialized) search engines, e.g. news, shopping, papers, recipes, reviews, etc.
* Creating huge dataset for analytics and machine learning
* monitoring sites (e.g price of an item or stock)
* comparing prices of different websites and providing deals to customers

**1.2 Web scraping vs web crawling**

The goal of such a web crawler is to learn what (almost) every webpage on the web is about and to mostly rank them on the basis of their content and metadata so that this can be indexed and retrieved fast when it's needed.

The same tools and techniques used for web crawling can be use for scraping where pieces of websites being crawled are extracted and stored for various application .

While the web crawler crawl websites with the purpose of indexing web pages for better information retrieval web scrapers while crawling extract portions of web pages into files or database.This data might then be used for various purposes

So web scraping has a more focused approach and crawling techniques such as focused crawling ,topical crawling are adopted while scraping

**1.3 How web crawler works**

The crawling process begins with a list of web addresses from past crawls and [sitemaps](https://support.google.com/webmasters/answer/156184?hl=en) provided by website owners. As crawlers visit these websites, they use links on those sites to discover other pages. The software pays special attention to new sites, changes to existing sites and dead links. Computer programs determine which sites to crawl, how often and how many pages to fetch from each site.

 Crawlers look at webpages and follow links on those pages, much like one would do you were browsing content on the web. they go from link to link and bring data about those webpages back to Google’s servers.

**1.3.1 Organizing information by indexing web pages**

When we search any term say “Web crawler “ in any search engine we get some results

Where did these results come from and why was this listed before the other one??

When we do a search, the search engine isn't actually going out into the world wide web, to run your search in real time. And that's because there's over a billion websites on the internet and hundreds more are being created every single minute. So if the search engine had to look through every single site to find the one you wanted it would just take forever.

So to make search faster, search engines are constantly scanning the web in advance to record the information that might help with search later. The search engine already has what it needs to give an answer in real time.

The Internet is a web of pages connected to each other by hyperlinks, search engines are constantly running a programme called a spider that crawls through these web pages to collect information about them. Each time it finds a hyperlink. It follows it until it has visited every relevant page, it can find on the entire internet for each page that spider visits, it records any information, it might need for a search by adding it to a special database called the search index.

When term is searched a search engine looks at each of those words in the search index to immediately get a list of all the pages on the internet, containing those words, but just looking for these search terms could return, millions of pages. So the search engine needs to be able to determine the best matches to show you first. the search engine uses its own algorithm to rank the pages based on what it thinks user wants want the search engines ranking algorithm might check if your search term shows up in the page title, they might check if all of the words show up next to each other, or any number of other calculations that help it better determine which pages user would want to see, and which you want Google invented the most famous algorithm for choosing the most relevant results for a search by taking into account how many other web pages linked to a given page. The idea is if lots of websites think that a

The Internet is constantly changing and expanding. Because it is not possible to know how many total webpages there are on the Internet, web crawler bots start from a seed, or a list of known URLs. They crawl the webpages at those URLs first. As they crawl those webpages, they will find hyperlinks to other URLs, and they add those to the list of pages to crawl next.

Given the vast number of webpages on the Internet that could be indexed for search, this process could go on almost indefinitely. However, a web crawler will follow certain policies that make it more selective about which pages to crawl, in what order to crawl them, and how often they should crawl them again to check for content updates.

1. **Social media**

Concept of social media has come from the need of humans to connect with people of common interests. e.g keeping up with friends from school,college or workplaces

Social networking has been the study of social sciences,computer sciences,statistics and economics long before the advent of internet social media platforms

Facebook,linkedin and twiiter are popular social media websites

The experiment which was performed as part of this thesis paper was the study of scraping the social media

**Social Network Analysis**

Social network analysis (SNA) is the process of investigating social structures through the use of [networks](https://en.wikipedia.org/wiki/Network_theory) and [graph theory](https://en.wikipedia.org/wiki/Graph_theory).I t characterizes networked structures in terms of *nodes* (individual actors, people, or things within the network) and the *ties*, *edges*, or *links* (relationships or interactions) that connect them

**6 degrees of separation**



Facebook ran the numbers and concluded that we are all much closer than the traditional “six degrees of separation.”The social media giant released [a report on its blog](https://research.facebook.com/blog/three-and-a-half-degrees-of-separation/) Thursday announcing “each person in the world” is separated from every other by “an average of three and a half other people.”

**Application of SNA**

Social network analysis has also been applied to understanding online behavior by individuals, organizations, and between websites.[[15]](https://en.wikipedia.org/wiki/Social_network_analysis#cite_note-Ghanbarnejad-15) [Hyperlink](https://en.wikipedia.org/wiki/Hyperlink) analysis can be used to analyze the connections between [websites](https://en.wikipedia.org/wiki/Website) or [webpages](https://en.wikipedia.org/wiki/Web_page) to examine how information flows as individuals navigate the web.[[65]](https://en.wikipedia.org/wiki/Social_network_analysis#cite_note-65) The connections between organizations has been analyzed via hyperlink analysis to examine

**Social media analytics**

Social media analytics is the process of gathering and analyzing data from [social networks](https://en.wikipedia.org/wiki/Social_network_service) such as [Facebook](https://en.wikipedia.org/wiki/Facebook), [Instagram](https://en.wikipedia.org/wiki/Instagram), [LinkedIn](https://en.wikipedia.org/wiki/LinkedIn) and [Twitter](https://en.wikipedia.org/wiki/Twitter). It is commonly used by marketers to track online conversations about products and companies. One author defined it as "the art and science of extracting valuable hidden insights from vast amounts of semi-structured and unstructured social media data to enable informed and insightful decision making

**3.Designing a web crawler**

**3.1 Design goals of a web crawler**

* **Coverage and completeness**

Coverage measures the relative number of pages discovered by the web crawler. Ideally given enough time the web crawler has to find all pages and build the complete model of the application. **This property is referred to as Completeness.**

* **Freshness**

The search engine index has to be updated constantly to reflect changes in web pages created dynamically. The ability of the web crawler to retrieve latest update sis measured through freshness.The crawler has to determine when and how it will pull for the changes in the crawled site

* **Politeness**

An important and old issue in designing web crawlers is called politeness Early web crawlers had no mechanism to stop them from bombing a server with many requests. As the result while crawling a website they could have launched an Denial of Service(DoS) attack unintentionally A barrage of requests in short order is considered “impolite”. So delays mut be introduced between requests.This especially has to be maintained by the URL distributor among concurrent or distributed crawlers which uses so that there are no redundant requests

* **Black-lists and traps**

Traps are seemingly large set of websites with arbitrary data that are meant to waste the web crawler paresources.These ae mostly spam sites designed to get backlinks to them to improve ranking Integration of black-lists will web crawlers to avoid traps

* **Scalability**

The web has been growing and evolving exponentially.So web crawlers have to keep up with those changes and still produce correct data

* **Efficiency**

While the web crawler is not supposed to crawl all pages and the process of gathering new data and update existingis a dynamic process its must have technique to do so in minimum amout of time otherwise information retrieved may be stale and of no use

**3.2 Challenges in designing a crawler**

The challenges one face while designing a crawler can be summed up by the following questions.The solutions to the queries has to be found keeping in mind the above design principles

1. **What pages should the crawler download?**

In most cases, the crawler cannot download all pages on the Web. Even the most comprehensive search engine currently indexes a small fraction of the entire Web .Given this fact, it is important for the crawler to carefully select the pages and to visit important pages first, so that the fraction of the Web that is visited (and kept up-to-date) is more meaningful while still

**2. How often should the crawler refresh pages?**

Once the crawler has downloaded a significant number of pages, it has to start revisiting the downloaded pages in order to detect changes and refresh the downloaded collection. Because Web pages are changing at very different rates the crawler needs to carefully decide which pages to revisit and which pages to skip in order to achieve high “freshness” of

pages.

1. **How should the load on the visited Web sites be minimized?**

When the crawler collects pages from the Web, it consumes resources belonging to other organizations Therefore, the crawler should minimize its impact on these resources (politeness)

4. **How should the crawling process be parallelized?**

Due to the enormous size of the Web, crawlers often run on multiple machines and download pages in parallel large number of pages in a reasonable amount of time. Clearly these parallel crawlers should be coordinated properly, so that different crawlers do not visit the same Web

site multiple times. However, this coordination can incur significant communication

overhead, limiting the number of simultaneous crawlers.

**4 Key factors for optimized crawler design**

Having discussed design principles we identified the following aspects were identified which would help us an optimized solution

**4.1 Page Selection**

Not all pages are of interest to the search .If the information retrieved is of a particular topic then relevant pages should be selected as early as possible

Given a web page p the importance of a page P , I(P) can be defined in the following ways

* **Similarity to driving query *:*** Since the query is the driving factor for the the similarity with it determines I(P) .TF-IDF is one of the techniques that may be use
* **Backlink count***:* Intuitively, a page p that is linked to by many pages is more important than one that is seldom referenced. On the Web, IB(p) is useful for ranking query results, giving end-users pagesthat are more likely to be of general interest
* **PageRank***:*The IB(p) metric treats all links equally. Thus, a link from the CNN home page counts the same as a link from some individual’s home page. However,since the Yahoo home page is more important (it has a much higher IB count), it would make sense to value that link more highly. The PageRank backlink metric,IR(p), recursively defines the importance of a page to be the weighted sum of theimportance of the pages that have backlinks to p.
* **Forward Link Count***:* For completeness we may want to consider a metric IF(p) that

counts the number of links that emanate from p. Under this metric, a page with many

outgoing links is very valuable, since it may be a Web directory

**4.2 Parallelization of Crawlers and distributed crawlers**

**Advantages of parallel crawlers**

* **Scalability** :

It is quite obvious that the stupendous task of crawling a web cannot be done without multiple crawlers working parallel

* **Network-load dispersion**:

Multiple crawling processes of a parallel crawler may run at geographically distant locations. In this way, we can disperse the network load to multiple regions.

* **Network-load reduction**:

In addition to dispersing load, a parallel crawler may actually reducethe network load. Note that the downloaded pages may need to be transferred later to a central location,so that a central index can be built.

**Some design choices for parallel crawlers**

Parallel crawler can implement the following to reduce network load

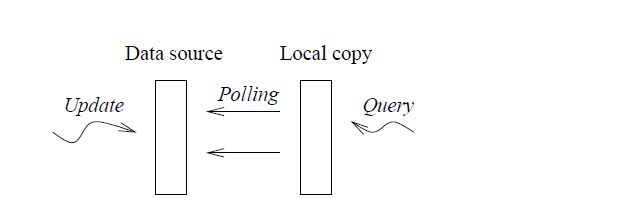
* **Overlap** : When multiple processes run in parallel to download pages, different processes may download the same page multiple times. One process may not be aware of the fact that another process has already downloaded a page
* **Compression:** Pages should be compressed before sending to central location
* **Difference:** Sending difference between previous image and current one instead of whole pages can save bandwidth
* Summarization: sending **data only to create central list**

,

The efficiency of parallel crawlers can be attained by considering few design choices as follows:

* ***Communication bandwidth****:* In order to prevent overlap, or to improve the quality the paraller crawlers must have optimized communication to reduce bandwidth

**4.3 Freshness and Page refresh policy**



* **Freshness**

Most of the web pages the crawler downloads on any given day were by it multiple times in the past. The crawler revisits them in order to keep the search engine’s index — and its search results — fresh.

Indeed, serving stale content such as YouTube pages with taken-down videos or failing to serve pages that the user knows to be relevant to her query due to new content is a cause of user dissatisfaction

Let S = {e1,....,eN} be the local database with N elements. Ideally,all N elements will be maintained up-to-date, but in practice, only M(< N) elements will be up-to-date at a specific time. (By up-to-date we mean that their values equal those of their real-world counterparts.) We define the fresh-ness of S at time t as F(S; t) = M/N.

Clearly, the freshness is the fraction of the local database that is up-to-date.

F(S; t) will be one if all local elements are up-to-date, and F(S; t) will be zero if all local elements are out-of-date.

The freshness of a local element ei at time t is

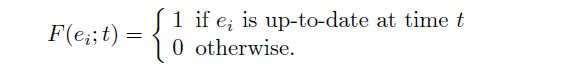
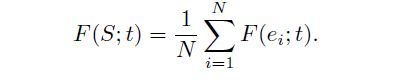


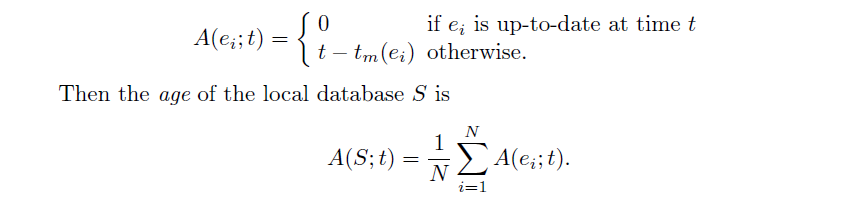
Fig 6

Then, the freshness of the local database S at time t is

****

**Fig 7**

* **Age :** To capture how old" a database is, we define the metric age as follows:

****

**Fig 8**

The age of S tells us the average age of the local database. For instance, if all real-world elements changed one day ago and we have not synchronized them since, A(S; t) is one day.

* 1. **Change frequency estimation**

For instance, popular news web sites, such as CNN and NY Times, update their contents periodically, whenever there are new developments. Also, many online stores update the price/availability of their products, depending on their inventory and on market conditions. Since the sources are updated autonomously, the clients usually do not know exactly when and how often the sources change. However, we believe that the clients can significantly benefit by estimating the change frequency of the sources.

**How do we trace the change of an element?**

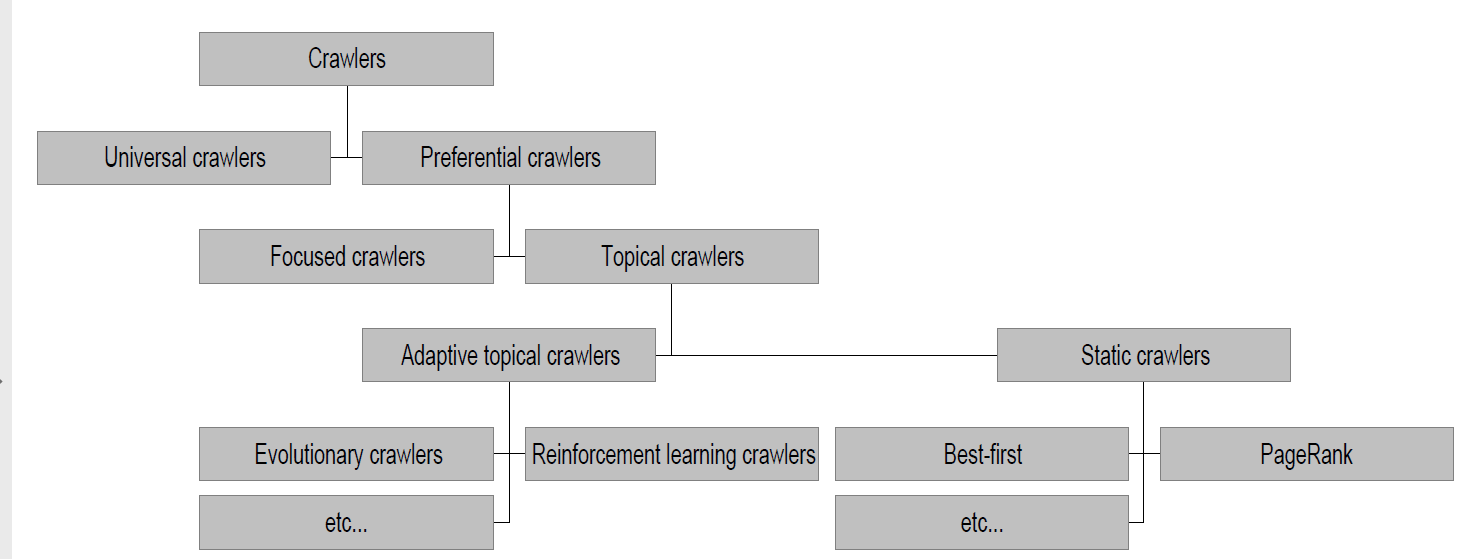
**Passive monitoring:**

We do not have any control over when and how often we access an element.In a web cache, for instance, web pages are accessed only when users access the page.Then acces to the element is random making it difficult to analyze

**Active monitoring:**

We actively monitor the changes of an element and can control the access to the element. Accessing at a regular interval will help estimating the frequency of change and then change frequency after observing the patter.For instance, if an element changes about once a day, it might be unnecessary to access the element every minutes, while it might be insufficient to access it every month.

1. **Types of crawler architecture**

****

**5.1 Universal crawler or Basic crawlers**

This is a sequential crawler Seeds can be any list of starting URLs Order of page visits is determined by frontier data structure .Stop criterion can be anything

**Universal crawler architecture**

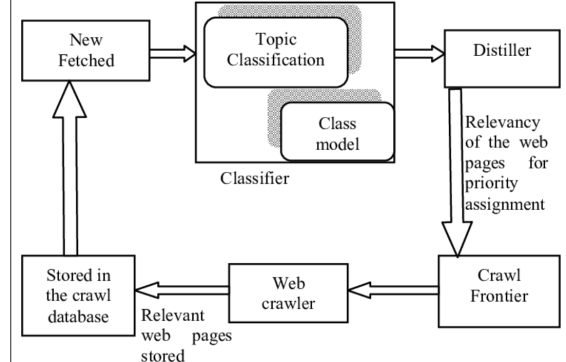
****

**5.2 Preferential crawlers**

Here the crawler have some selective bias and according to that importance I(P ) is calculated and maintained in a priority queue .This can be based on anything like closenees to seeds,most popular,largest pagerank,higes rate/amount of change in a website

5.3 **Focussed crawler** :

Fig 2.Focussed web crawler



A general purpose Web crawler gathers as many pages as it can from a particular set of URL’s, Where as a focused crawler is designed to only gather documents on a specific topic, thus reducing the amount of network traffic and download. The goal of the focused crawler is to selectively seek out pages that are relevant to a pre-defined set of topics.The topics are specified not using keywords, but using exemplary documents. Rather than collecting and indexing all accessible web documents to be able to answer all possible adhoc queries, a focused crawler analyzes its crawl boundary to find the links that are likely to be most relevant for the crawl, and avoids irrelevant regions of the web.

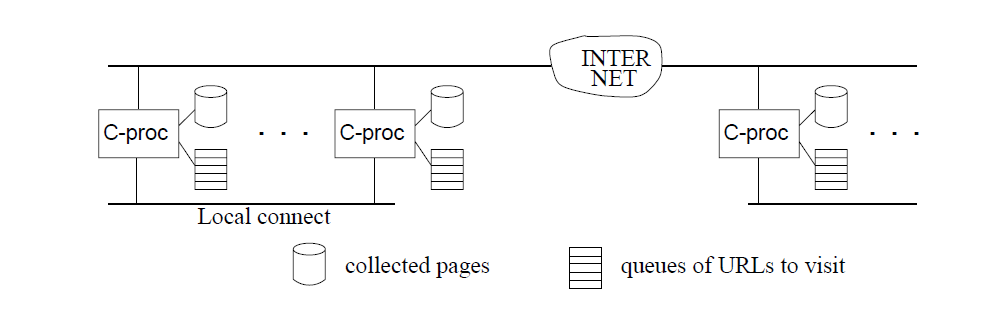
A focused crawler predicts the probability that an unvisited page will be relevant before actually downloading the page.Sometimes I use supervised learning techniques.This leads to significant savings in hardware and network resources, and helps keep the crawl more up-to-date. [5]

**5.4** **Topical crawler**

A topical web crawler is to collect web pages that describe some pre-specified topics. The web pages collected by the topical crawler share the same or similar words and however among them not a few pages can be irrelevant to the given topics. In particular, the performance of topical crawler degrades for a more specific topic. To achieve successful topical crawling, an additional job is required to actively filter out the pages irrelevant to the given topics. For this a machine learning architecture is used that can effectively handle not only literal term features but also numeric meta-features to improve topical web crawler; in our work we intend to more precisely crawl the web pages about ‘Jadavpur university IT department” as a specific topic. For the numeric meta-features logistic regression and random forest learning algorithms and for the literal word features, Naive Bayes and support vector learning algorithms. [6]

**5.5 Parallel crawler**

Multiple crawlers are often run in parallel, which are referred as Parallel crawlers . A parallel crawler consists of multiple crawling Processes called as C-procs which can run on network of workstations . The Parallel crawlers depend on Page freshness and Page Selection .Parallelization of crawling system is very vital from the point of view of downloading documents in a reasonable amount of time .

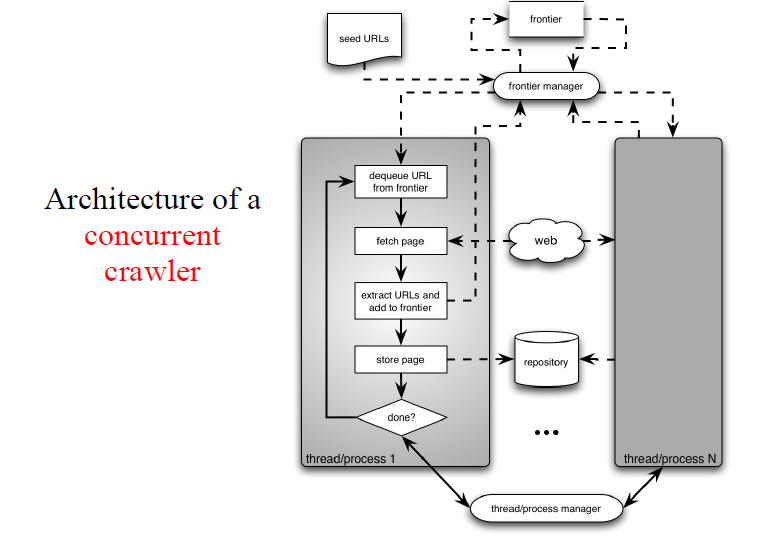


A Parallel crawler can be of the following types

1. **Intra Site crawler** : When all C-proc’s run on the same local network and communicate through a high speed interconnect (such as LAN), we call it an *intrasiteparallel crawler*
2. **Distributed crawler:**When a crawler’s C-proc’s run at geographically distant locations

connected by the Internet (or a wide area network). When C-proc’s run at distant locations and communicate through the Internet, it becomes important how often and how much C-proc’s need to communicate

4.2.4 **Concurrent crawlers**

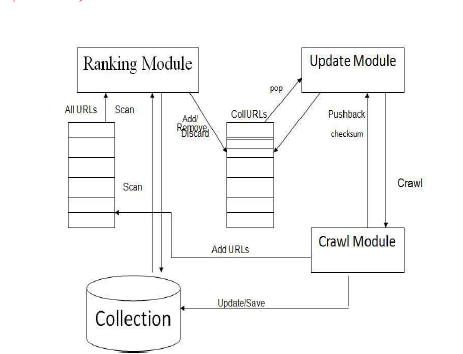


Can use multi-processing or multi-threading .Each process or thread works like a sequential crawler, except they share data structures: frontier and repository

• Shared data structures must be

**4.5 Incremental crawler**

A traditional crawler, in order to refresh its collection, periodically replaces the old documents with the newly downloaded documents. On the contrary, an incremental crawler incrementally refreshes the existing collection of pages by visiting them frequently; based upon the estimate as to how often pages change [ It also exchanges less important pages by new and more important pages. It resolves the problem of the freshness of the pages. The benefit of incremental crawler is that only the valuable data is provided to the user, thus network bandwidth is saved and data enrichment is achieved.Also politeness of crawler can be maintained



6 **Crawler algorithms**

7 **Politeness and Robots.txt file**

A robots.txt file is a text file that lives on a web server and specifies the rules for any bots accessing the hosted website or application. These rules define which pages the bots can and can't crawl, which links they should and shouldn't follow, and other requirements for bot behavior.Search engines have two main jobs:

1. Crawling the web to discover content;
2. Indexing that content so that it can be served up to searchers who are looking for information.

After arriving at a website but before spidering it, the search crawler will look for a robots.txt file. If it finds one, the crawler will read that file first before continuing through the page. Because the robots.txt file contains information about *how* the search engine should crawl, the information found there will instruct further crawler action on this particular site. If the robots.txt file does *not*contain any directives that disallow a user-agent’s activity (or if the site doesn’t have a robots.txt file), it will proceed to crawl other information on the site.

Here are the following follow actions in robots.txt

##### **Blocking all web crawlers from all content :** User-agent: \* Disallow:

##### **Blocking a specific web crawler from a specific folder :**

User-agent: Googlebot Disallow: /example-subfolder/

* **Inserting a delay for a specific web crawler :**

User-agent: Bingbot

Crawl-Delay :120

This introduces crawl delay of 120 msec begore crawling any page

1. **Implementation challenges**

**Requests Getting Blocked**

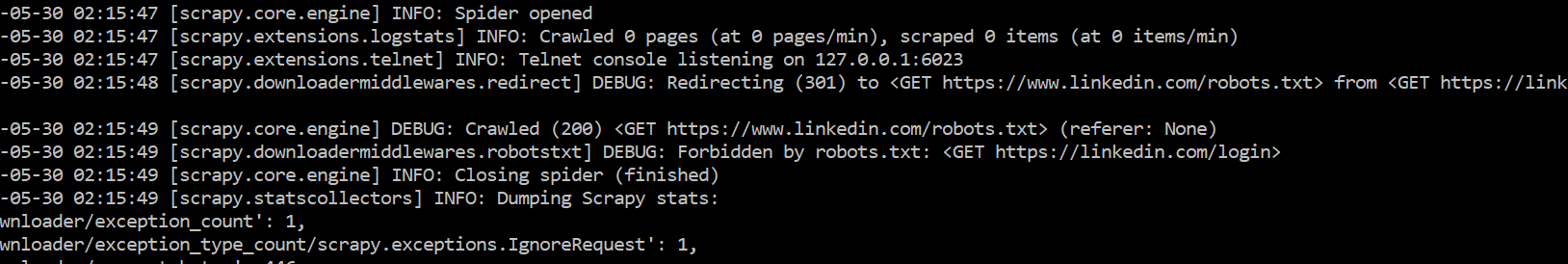
Apart from the design the first and major challenges in scraping data that was faced in crawling any website is justifiably the issue of access .

Most popular websites like Facebook,Linkedin do not allow scraping or crawling for many reasons

* Respecting privacy of users
* preventing contents from being crawled as not all information is for public domain .Any organization may have some confidential inforation
* In many countries there are laws preventing exposure of sensitive private information to public domain which attract heavy penalties if not adhered to/

Robots.txt policy

Linkedin websites have defined **User-agent**: \* or **Disallow:/** in robots.txt So any automated login and scraping is generally forbidden .So whenever there request that try to access webpages that are include in robots.txt it immediately forbods

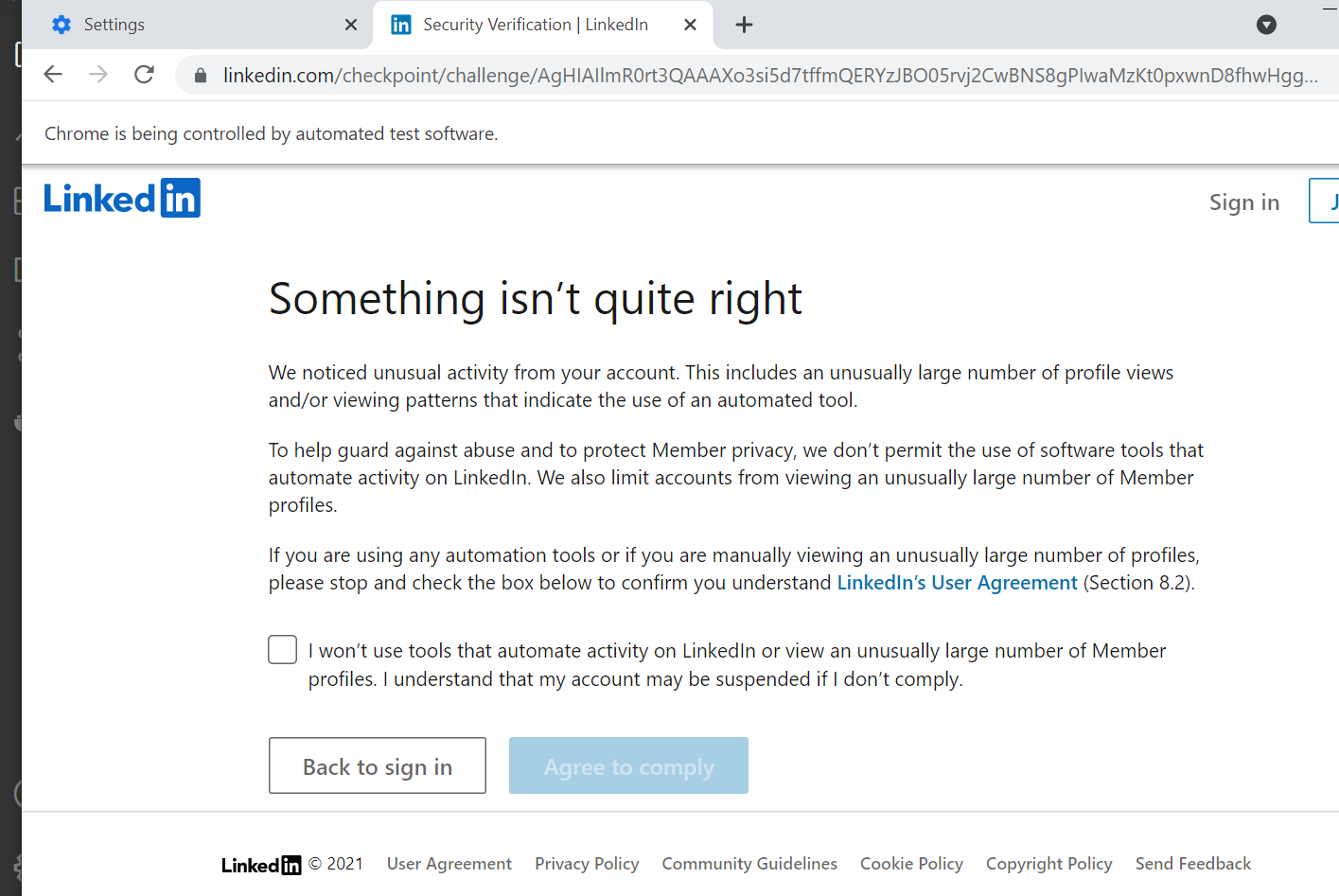


As we see here the attempt to login using python request is getting redirected to robots.txt and get forbidden http status (403) .

**Rate Limiting**

Rate limiting is a strategy for limiting network traffic. It puts a cap on how often someone can repeat an action within a certain timeframe .It is mostly done to prevent DDOS attacks

Mutiplle time during running tests we encountere following warning



* How to overcome this challenge then

1. Scraping refused

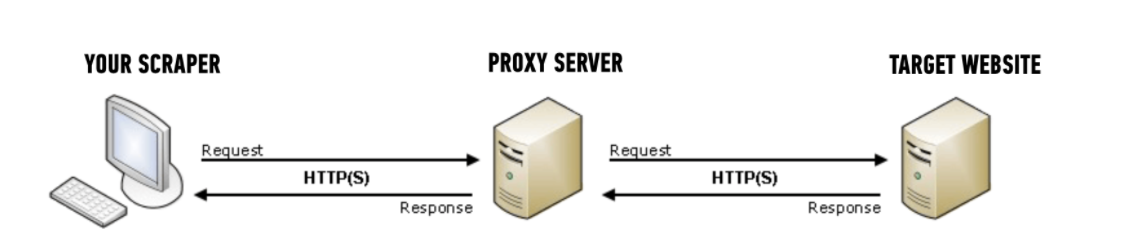
Most popular websites have software to detect huge number of requests from one IP address and blocking the ip suspecting DDOS attack also many website looks for user agents headers and if not found may reject

3 . **DNS resolution**

It is a well-known bottleneck in web crawling. Due to the distributed nature of the Domain Name Service, DNS resolution may entail multiple requests and round-trips across the internet, requiring seconds and sometimes even longer. Right away, this puts in jeopardy the goal of fetching several hundred documents a second. A standard remedy is to introduce caching: URLs for which we have recently performed DNS lookups are likely to be found in the DNS cache, avoiding the need to go to the DNS servers on the internet. However, obeying politeness constraints (see Section [20.2.3](https://nlp.stanford.edu/IR-book/html/htmledition/the-url-frontier-1.html#sec:frontier) ) limits the of cache hit rate.

**Solutions to prevent blocking:**

**Using Proxy server**



**Benefits of proxy server**

Many large sites have software in place to detect when there are a suspicious number of requests coming in from one IP address, since this usually indicates some sort of automated access – it could be scraping,

The way this rate limiting software is usually setup, if too many requests come in from one IP address in a short amount of time, then the site will return some sort of error message to “block” future requests from that client for a pre-set period of time.

* **Faking user-agents**

When a browser connects to a website, it includes a User-Agent field in its HTTP header. The contents of the user agent field vary from browser to browser. Each browser has its own, distinctive user agent. Essentially, a user agent is a way for a browser to say “Hi, I’m Mozilla Firefox on Windows” or “Hi, I’m google chrome on an iPhone” to a web server.

The web server can use this information to serve different web pages to different web browsers and different operating systems.

Many websites check this user agent and if not desired string value like

“Mozilla/5.0 (iPad; U; CPU OS 3\_2\_1 like Mac OS X; en-us) AppleWebKit/531.21.10 (KHTML, like Gecko) Mobile/7B405” is not found then denies service .

So fake user-agents may be used with rotating user-agents so that it will appear to the website that different client request are coming

• URL canonicalization

**7 Test experiment :**

**Social media crawling/scraping**

Social media being hugely popular presents wealth of information about users who voluntarily provide many data point which help us to profile them .Here people along with thei basic information provides their preference and expectations through their likes and dislikes(through playful emojis).

This data is worth mining as there are many consumers of the data e.g a college department may be interested in where alumni is working,doing research as they may need to connect with alumin for various purposes.There can be other interests like connecting with potential employes,employee o,investors,buyers

**Why Linkedin**  **??**

The experiment was done on linkedin as first of all its popular website used by serious professionals. So mostly the data can be considered as genuine as people put up profiles to connect with batchmates,colleagues and to expand their professional network connect with potential recruiters,clients or marketeers

Also the web elements were such that they can be scraped

**Why the experiment couldn’t be done on facebook??**

Had the experiment be done on facebook it would have got more data as it more popular but Search strings like “graduated from jadavpur university “ only returned groups like “Jadavpur university” that required membership to see member.Facebooks seems to protect privacy of users more somehow.Not that it was impossible .

One way would starting with some seed user who is a known alumni and then jumping to profiles from contacts and so on scarping info could b done

**Objective of the crawler :**

To find alumni of any institute (focused crawler) to know may be questions like

* Who are they??
* Which year they pass out?
* Which Branch they belonged to??
* Where are they now??
* Where are they working now??
* Are they doing research??
* What knd of research are they doing or what fields they are working ??
* What organizations are they working for ??
* What type of roles and profiles are they working

**Test scraper**

**Technologies used** :Python,BeautifulSoup(scraping library python),Selenium(testing automation software) scrappy

**BeautifulSoup**

BeautifulSoup the open source python library was very useful .Its widely used to parse html and xml web pages .It provides many function to navigate DOM vertically (parent and child),horizontally(siblings)

To parse a document simply pass it into the BeautifulSoup constructor. You can pass in a string or an open filehandle:

html\_soup = BeautifulSoup(browser.page\_source,'html.parser')

    name = html\_soup.find("h1", {"class": "text-heading-xlarge inline t-24 v-align-middle break-words"})

Some functions like find() and find\_all() can be done with any pattern like text,class\_name ,id or tag nme element or by the type of element (like div,hred etc) like this which used to find all URLS within a pages <a> tag

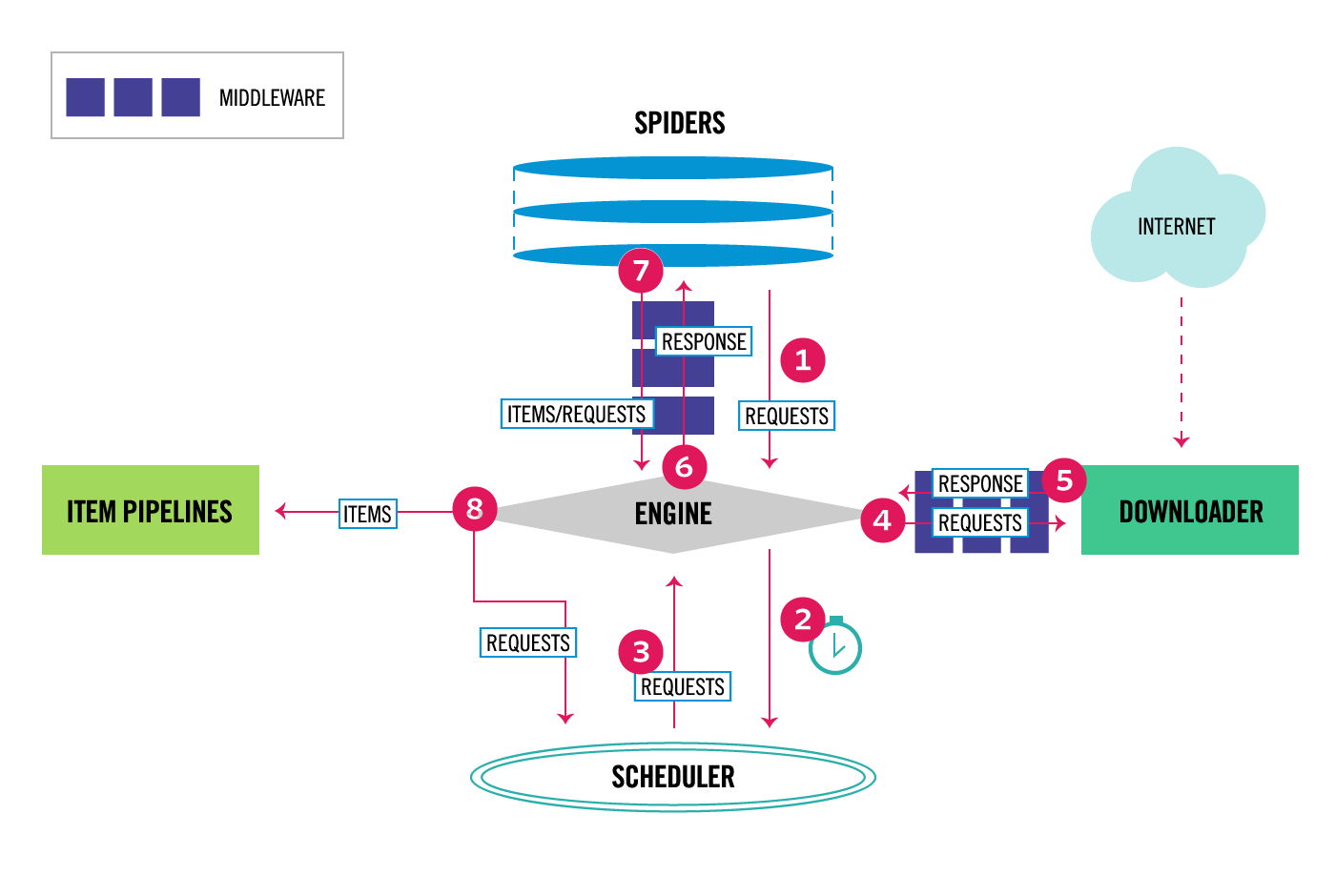
**for** link **in** soup.find\_all('a'):

**print(link.get('href'))**

**Scraping with Scrapy**

scrapy is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [web-crawling](https://en.wikipedia.org/wiki/Web_crawler) [framework](https://en.wikipedia.org/wiki/Web_framework) written in Python. Originally designed for web scraping, it can also be used to extract data using [APIs](https://en.wikipedia.org/wiki/Application_programming_interface) or as a general-purpose web crawler

**Architecture of scrapy**

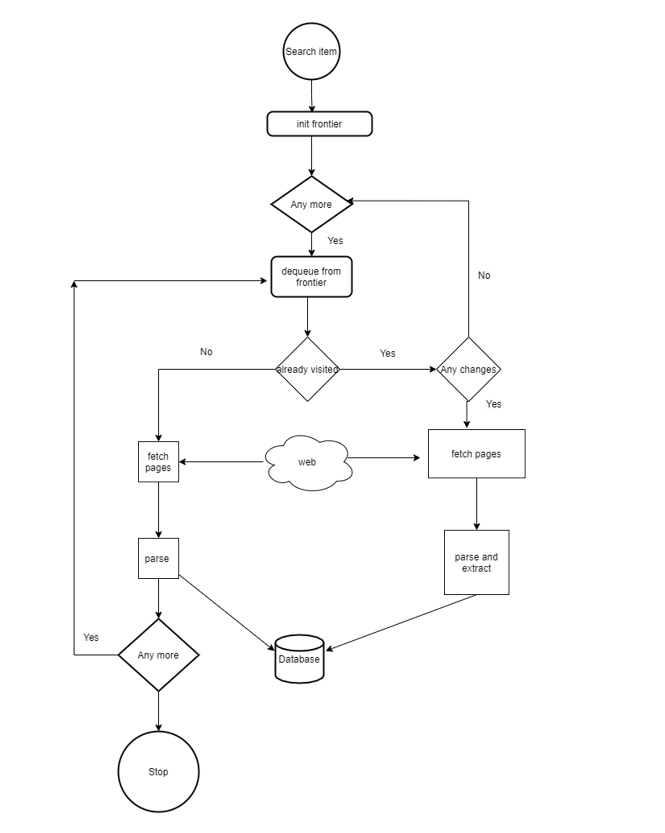
****

**Advantages of scrapy**

Requests are [scheduled and processed asynchronously](https://docs.scrapy.org/en/latest/topics/architecture.html#topics-architecture). This means that Scrapy doesn’t need to wait for a request to be finished and processed, it can send another request or do other things in the meantime. This also means that other requests can keep going even if some request fails or an error happens while handling it.

While this enables you to do very fast crawls (sending multiple concurrent requests at the same time, in a fault-tolerant way) Scrapy also gives you control over the politeness of the crawl through [a few settings](https://docs.scrapy.org/en/latest/topics/settings.html#topics-settings-ref). One can do things like setting a download delay between each request, limiting amount of concurrent requests per domain or per IP, and even [using an auto-throttling extension](https://docs.scrapy.org/en/latest/topics/autothrottle.html#topics-autothrottle) that tries to figure out these automatically.

**7.2 Flowchart**



While (There are items to crawl) :

Fetch

**7.3 Crawling Strategies attempted**

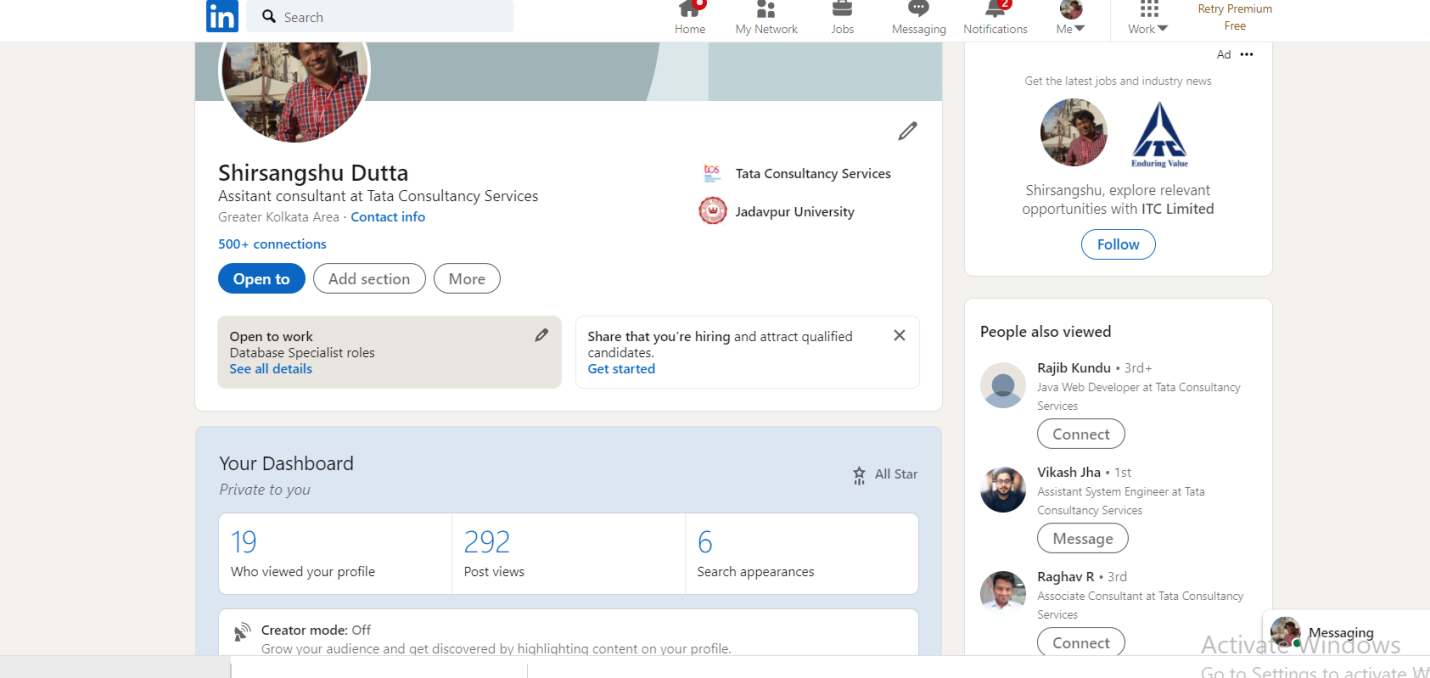
**Approach 1:**

**Visiting a profile and looking up all the profiles in connection**

Most profiles have connections which are of almost similar profiles like may be graduated from same college or working in same company .Not only these linkedin recommends similar profile which may not be in conncetion(or friends) but have similarty wrt any feature like roles,company,institute or research area

So this was a good way looking for alumni of “Jadavpur university ” looking up profiles from

“People also viewed” section starting from a seed profile



Advantages

This section shows profile which may be secondary or tertiary or more connections which linkedin has already computed .So all the similarities which we may have otherwise have to be found are already done by algorithms of linkedin and put at one place

Disadvantage :

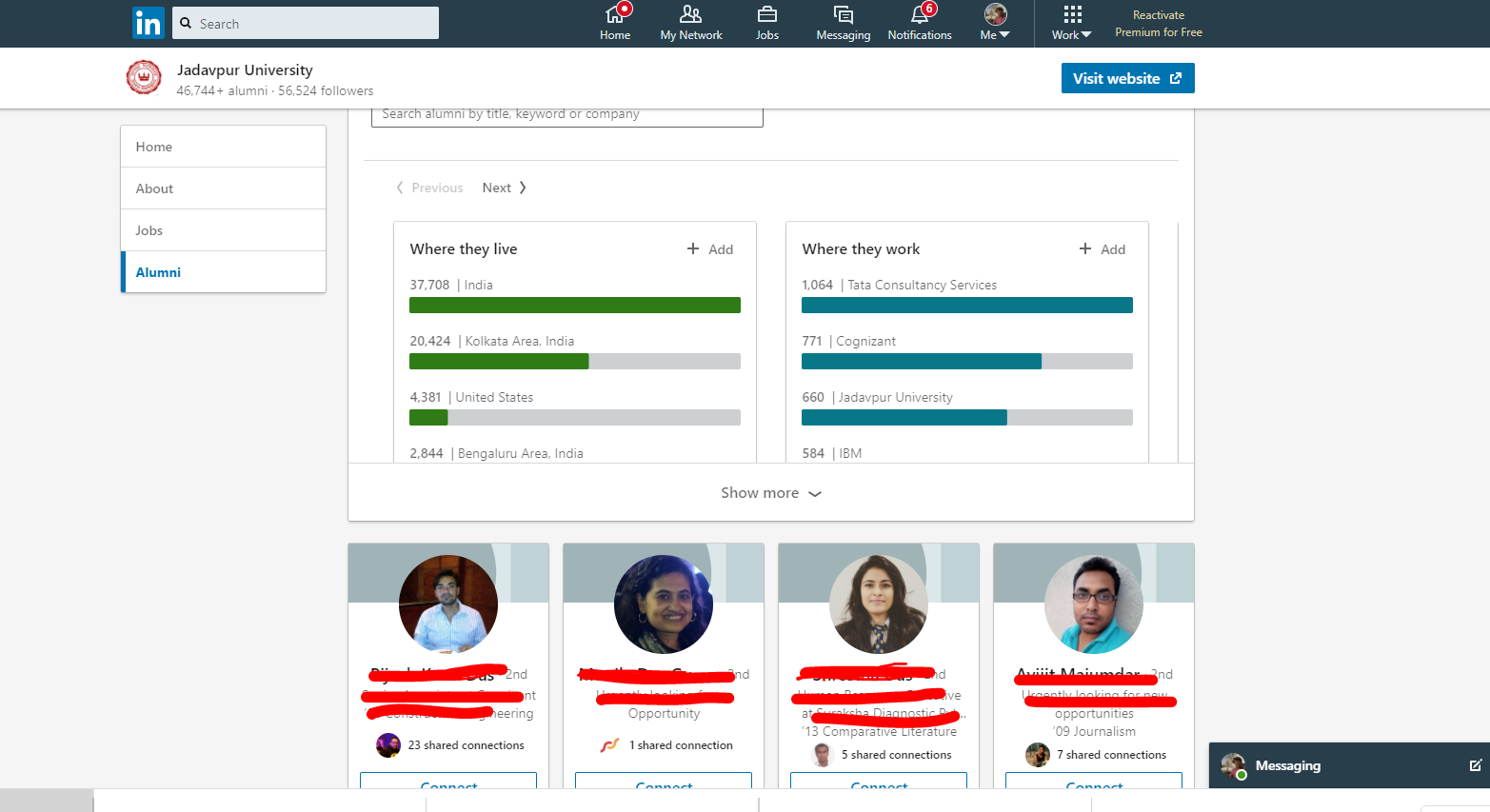
As mentioned this section shows profile based on a various feature like company worked,current roles so if we are doing a search for e.g we want alumni of institute we may have to come across profiles irrelevant to our subject and waste time looping through them

**Approach 2:**

**Looking up alumni public groups**

Many alumni are member of public groups of their alma mater .So this will be a spot where directly lot of alumni information can be collected right away

Disadvantages : Many alumni may not be part of this group and hence their infor not vaialble



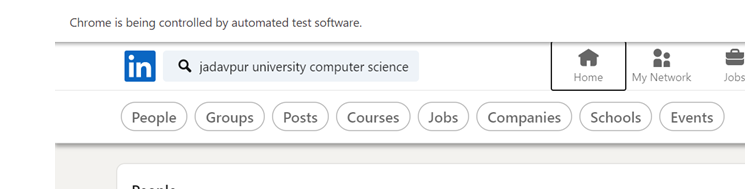
**Advantages :** Since the required profils are already clustered together through public group the extra effort and time of information retrieval can be save

**Approach 3 :**

This approach was the easiest and optimal in quality of data it returned and ease of navigating through the profiles

**Search the search box with target query**

Here we invoke a search query in the universal search box of linked in app

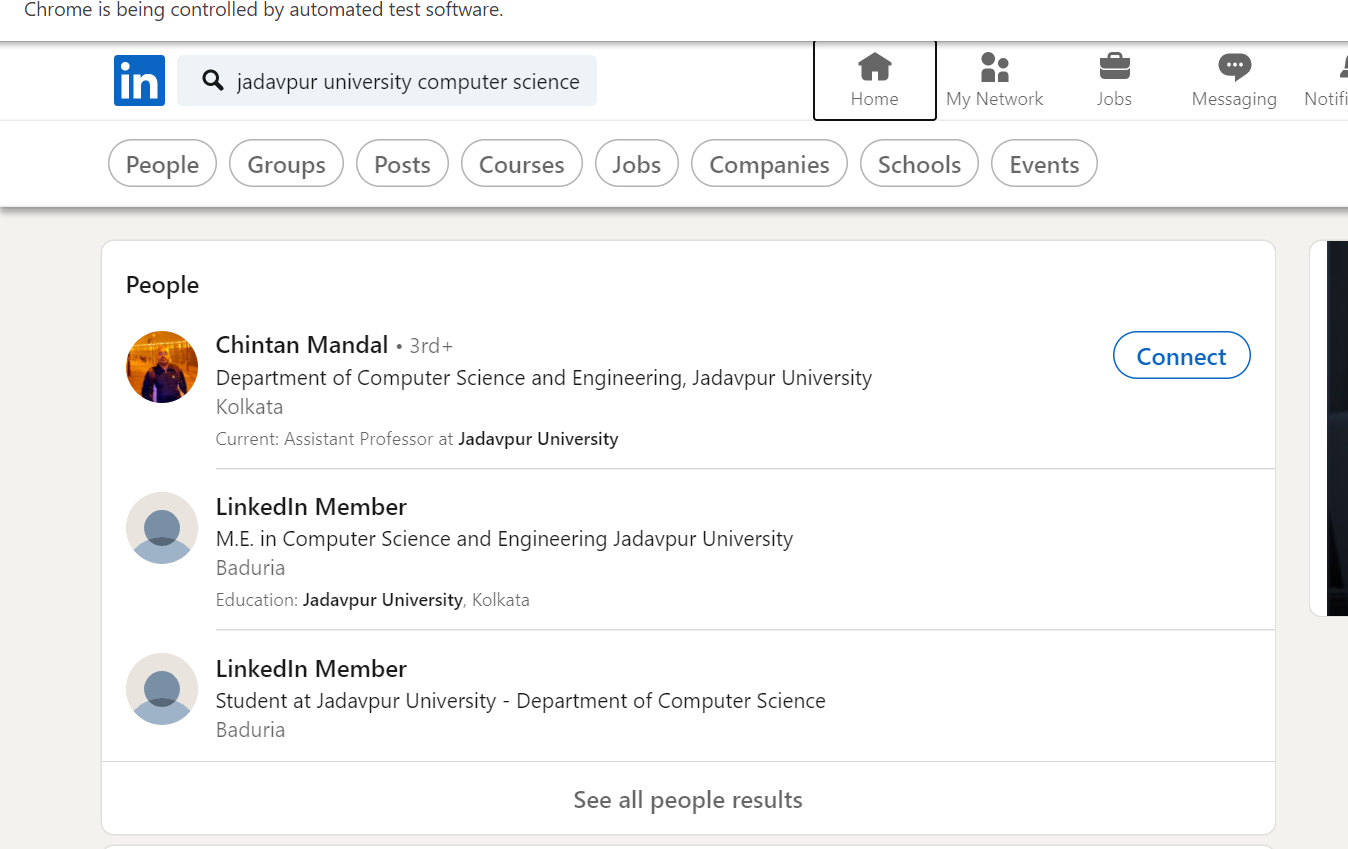


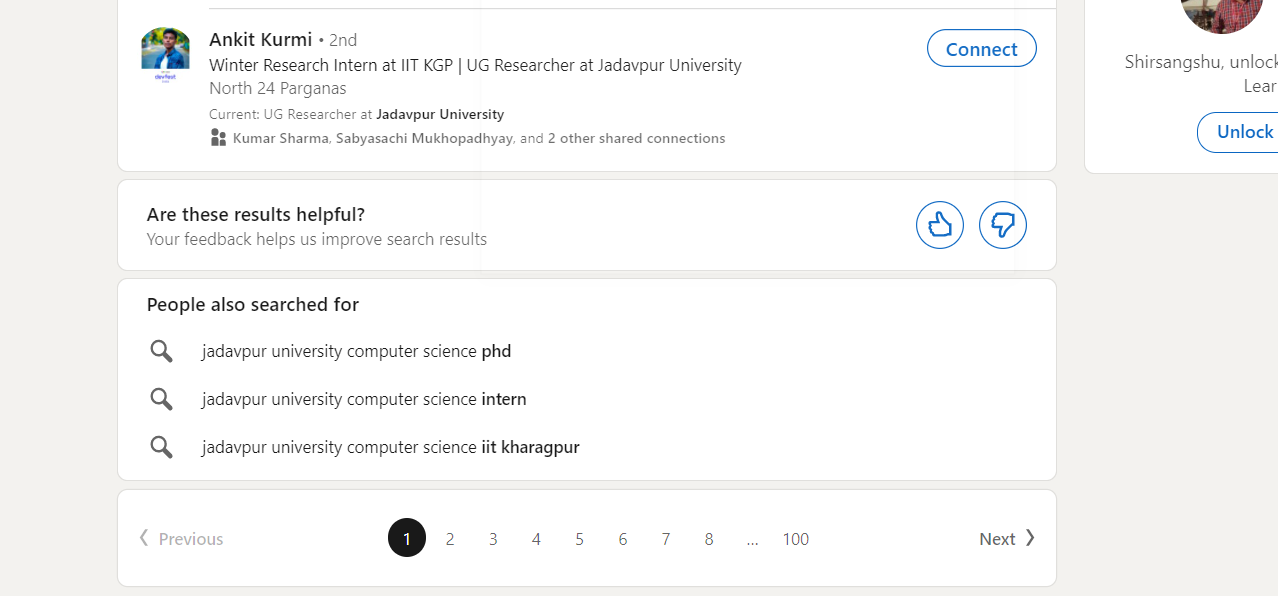
**Steps in the solution**

1. Automatic logging using Selenium

2 .Invoke a search by keying in the keywords for e.g “Jadavpur University Computer Science”

get the results like .There are many pages of output like this .





1. We get the DOM structure using developer tools and search for common patterns which represents each profile in the search results like div class “**entity-result\_\_content entity-result\_\_divider pt3 pb3 t-12 t-black—light”**



1. Collect all the links to be visited in a list while looping through all the result pages
2. A list of all scraped profiles is maintained .So links already visited are not repeated to save resources and make the crawler fast.In this way also the number of request to the website can also be reduced which will help the crawler not to get blocked.

6. Now once the frontier list is prepared loop and visit each profile and collect the relevant data like name of alumni , department ,year of passing ,course attendedexperience details like current company working for ,Experience,current roles and responsibilities

7. Once the data is created its dumped in an excel file and inserted into a database

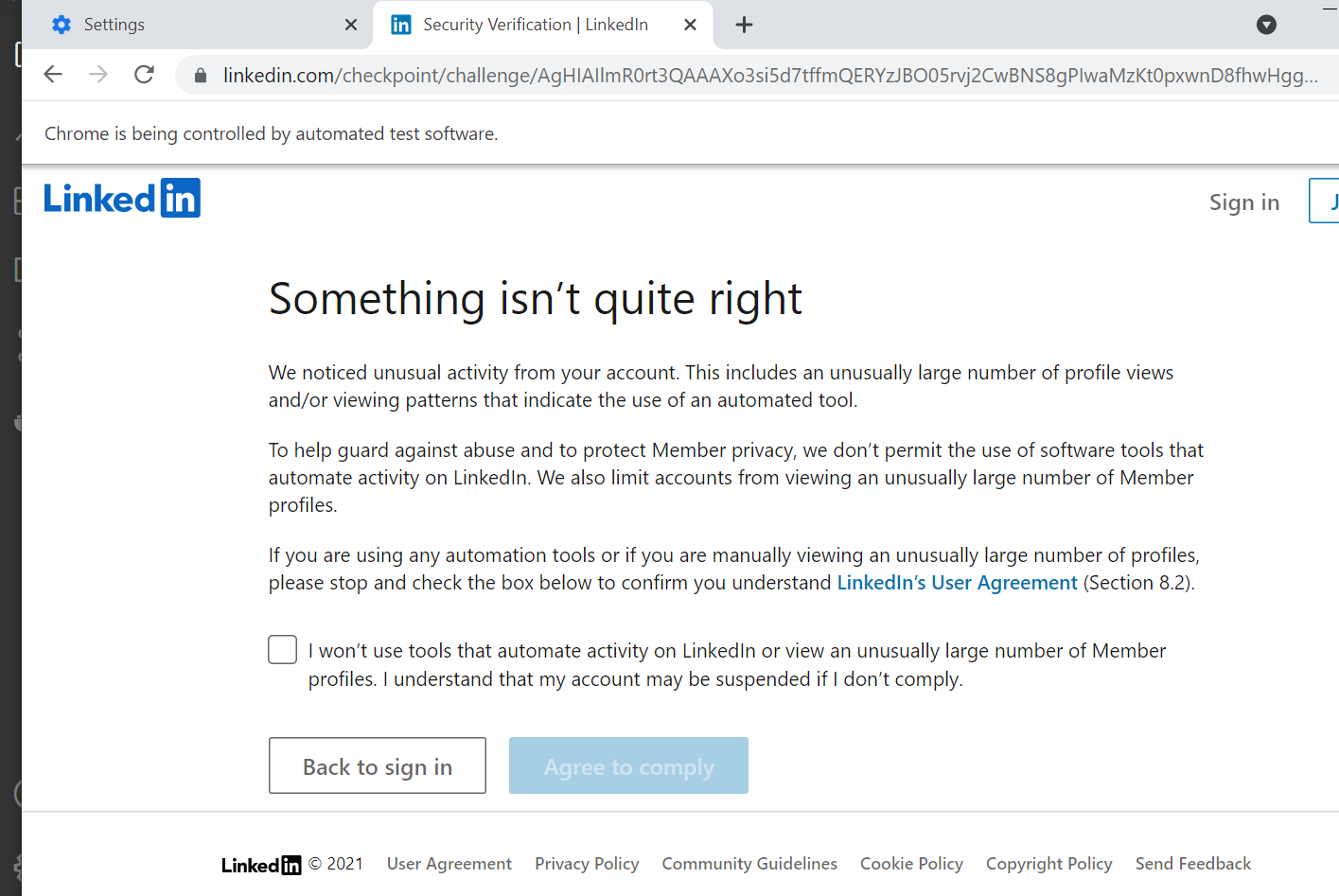
**7.4 Steps of crawling**

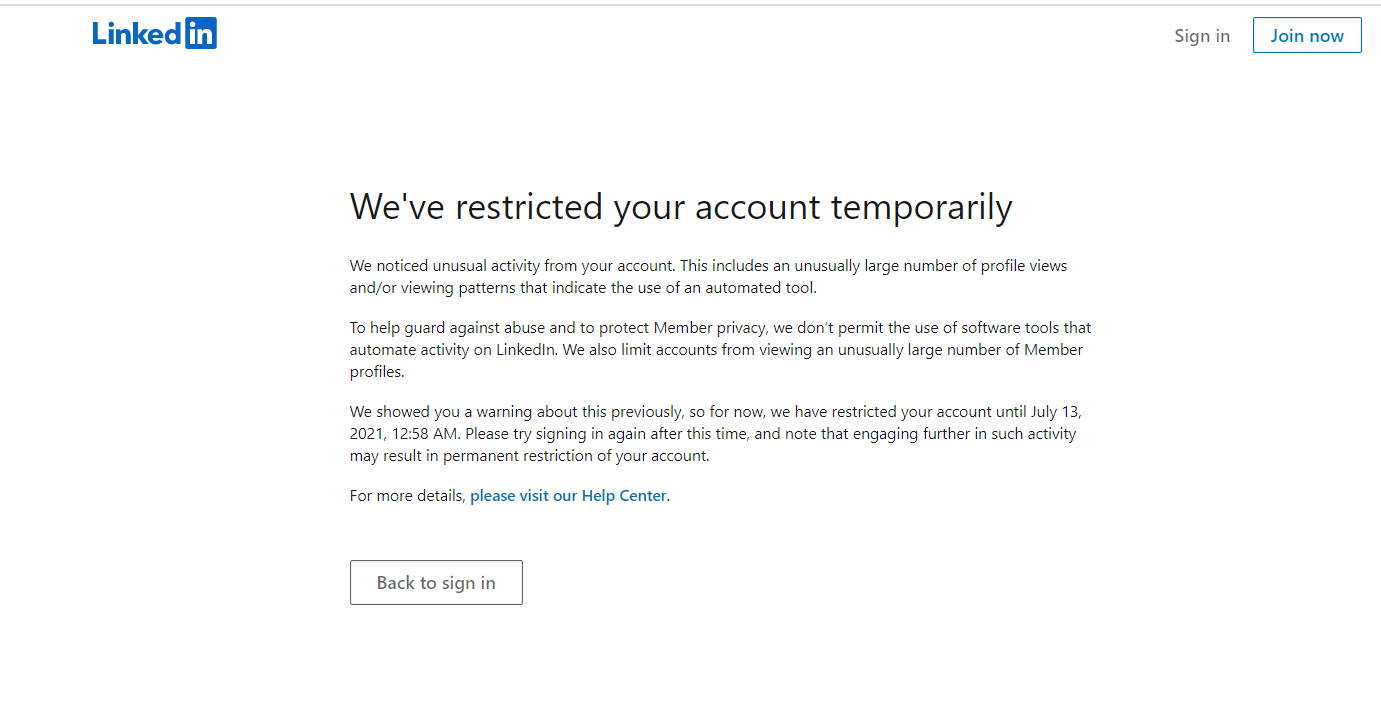
1. Problems in login :

At first login was attempted using python requests library .It was getting http 999 (requet denied) status .So had to use selenium to emulate web browser.

1. Intially a steady crawler was used i.e the crawler would visit,parse and store everytime it runs

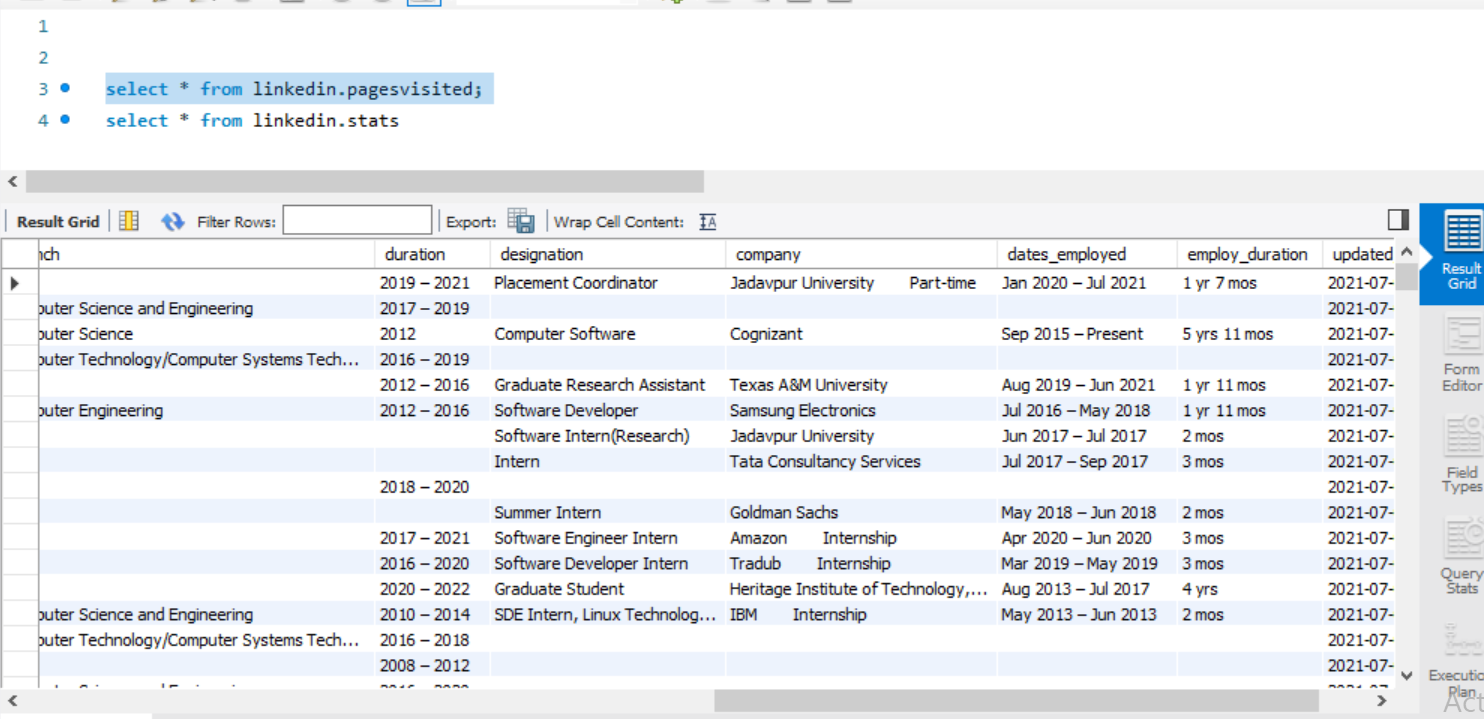
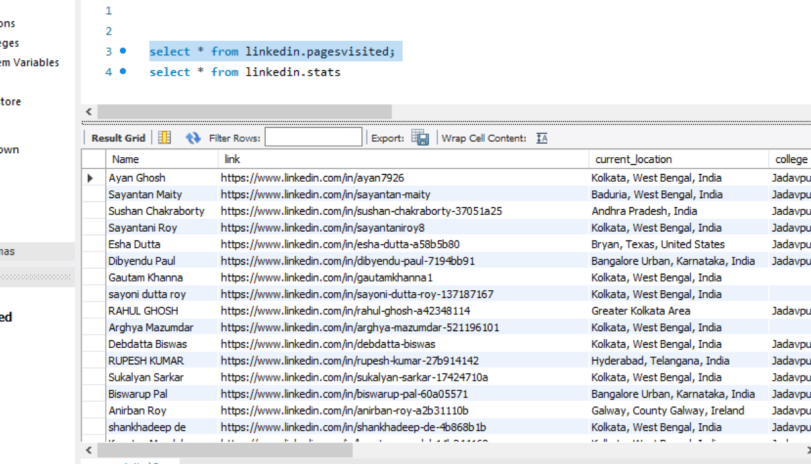
While this got the fresh data everytime it was very slow having to revisit profiles mostly which have not changed at all.Also may be linkedin webmaster were detecting pattern of huge hits from same ip in the same session and issued warnings like this one .





Also there were other issues such as interruption of network,crawler getting hung

1. To solve this crawler was made incremental .In this way the no of requests can be controlled few at a time .The profiles that have been visited and data retrieved from them are saved .When the crawler runs each time it first looks up the list and if already covered does not revisit them .It makes a frontier of only the new profiles that appear in the crawl and visit them to scrape the profile.
2. All the data in the incremental frontier that has been scraped are then posted to a database after each iteration



1. All the statistics like total profiles scraped in a run,time taken in a session is captured and saved to db

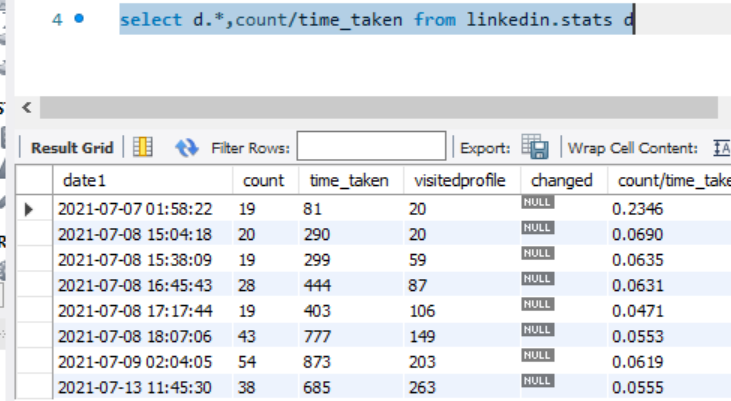
**7.5 Change monitor program**

Periodicallly batch runs to checks for updates of all the profiles scraped and if there is an update it applies the update and updates the profile changes in the following way

1. From a queue of visited profiles it revisits one by one (though not more than 20 at a time because that may lead to being blocked)
2. It then compares with stored data in database
3. If unchanged skips to next profile else updated the data with latest changes

**Metrics collected from experiment**

Several runs were done and the following metrics were collected .



**Total profiles scraped** : 263

**Time taken** : 3852 seconds (1.07 hrs)

From the data we around 263 profiles were scraped in about one hour

The scraping rate was almost near about the same even though it was done on various dayd

To be noted that this includes delay of a random wait time of 1-3 seconds between scraping of each profile which was inserted for the sake of politeness and avoid getting blocked.So actual time taken is actually < 1 hr

* 1. **Analyzing the data collected**

***“Data is not useful until it becomes information”***

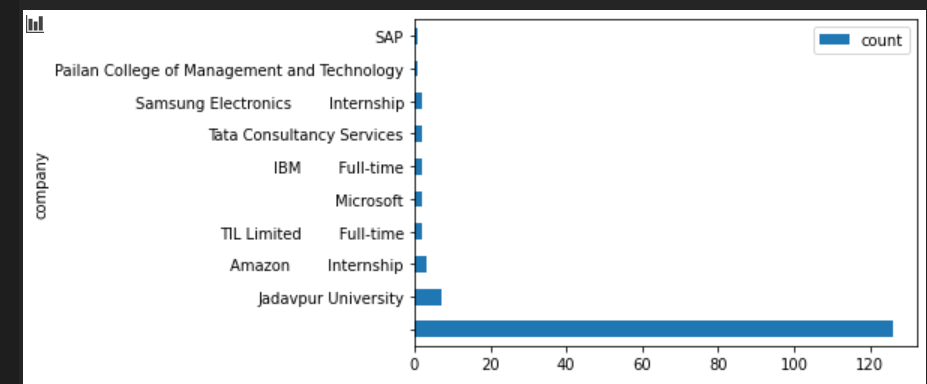
Let us looked into the data collected and letus try to derive some assumption and inferences from the data collected

***“A picture is worth a thousand words”***

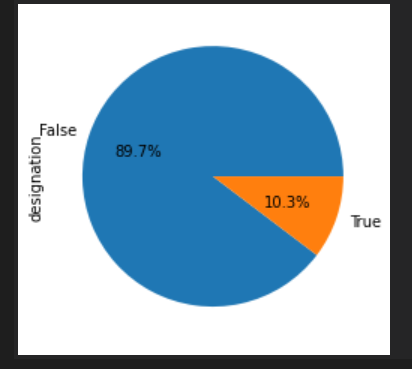
So the data is presented in various common data visualization which light up many inferences

* Where are they working now ??

**Top 10 companies where alumi are working**

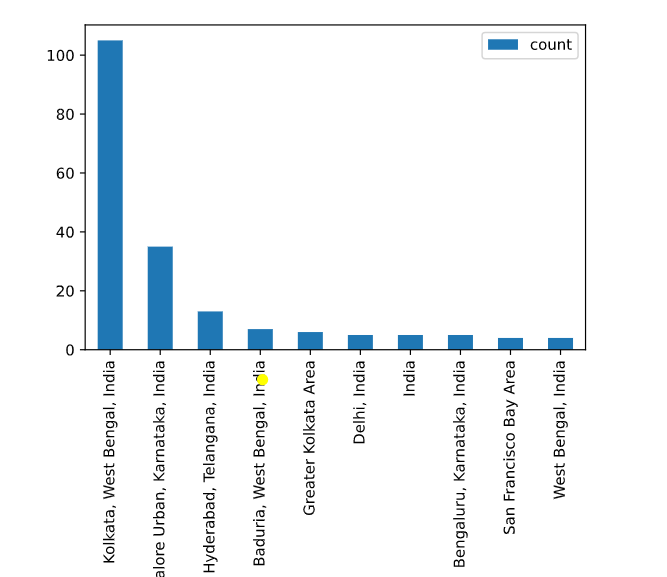


* Q. How many among alumni pursuing research careers



* Q. What are the major roles the are working in

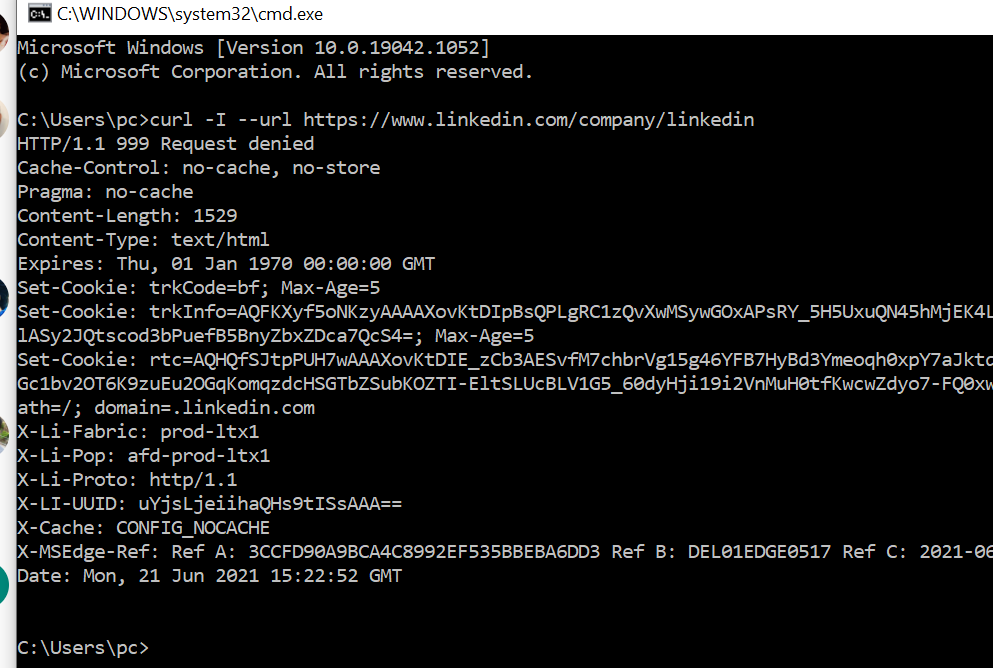


* q. Where do alumni live (Top 10 locations)
* 

**Challenges faced**

Request denied (999 status)

A sample HEAD request:



For this selenium webdriver was used which emulates a browser.The response was definitely slower but

* **Non uniform profiles**

Grossly most profiles are non uniform . This is peculiarity of social media where no two profiles are\ same The length of a profile may not be same .

Some profile may have put a resume or som eresearch paper so I does not fit into template used for searching.

The root cause is the whole page is no getting loaded all the time for longer profiles.So the html file that is render may not contain all the web element s or scraping the required info.

Some profiles may educational details but experience details are not there which return none so for the program to be resilient that had to be handled

* Some dynamic rendering is there using Javascipt which cannot be scraped
* Many have kept their profiles private so thye are simply showing as “Linkedin Member”

These profiles although visible in search results are thus not worth scraping as no usable information is gathered from them

* **Changing web element properties**

Linkedin keep changing the name of elements or class of web elements from time to time So one approach of extracting a particular div of class name “education-section” may not work few days later

* **Slowness of program**

The program ran in a single thread looking up profiles one by one . So it consumed time more so because of the delays introduced for politeness and to prevent from getting blocked.

* **Freshness of data**

Profiles keep changing as people change their education or experience info.To achieve this

the profiles that are already scraped in one go needed to be scraped again.There are no other ways to determine change .Search had to scheduled. But that posed a threat of being detected for visiting same links again

Citations and References

* Social network theory (cf. Wasserman & Faust)
* Cho, J. and Garcia-Molina, H., 1999. *The evolution of the web and implications for an incremental crawler*. Stanford.
* 5.Focused crawling: a new approach to topic-specific Web resource discovery Soumen Chakrabarti a , Martin van den Berg b , Byron Dom c a Computer Science and Engineering, Indian Institute of Technology, Bombay, 400076, India b FX Palo Alto Laboratory, 3400 Hillview Ave, Bldg 4, Palo Alto, CA 94304, USA c IBM Almaden Research Center, 650 Harry Rd, San Jose, CA 95120, USA
* 6. Tae Jun Kim and Han- Joon Kim, 2017. Machine Learning-Based Topical Web Crawler: An Ensemble Approach Incorporating Meta-Features. Journal of Engineering and Applied Sciences, 12: 4651-4656.

DOI: [10.36478/jeasci.2017.4651.4656](http://dx.doi.org/10.36478/jeasci.2017.4651.4656)