**CHAPTER 1**

**INTRODUCTION**

Web scraping, also known as web extraction or

harvesting, is a technique to extract data from the

World Wide Web (WWW) and save it to a ﬁle

system or database for later retrieval or analysis.

Commonly, web data is scrapped utilizing Hyper-

text Transfer Protocol (HTTP) or through a web

browser. This is accomplished either manually by

a user or automatically by a bot or web crawler.

Due to the fact that an enormous amount of het-

erogeneous data is constantly generated on the

WWW, web scraping is widely acknowledged as

an efﬁcient and powerful technique for collecting

big data (Mooney et al. 2015; Bar-Ilan 2001). To

adapt to a variety of scenarios, current web scrap-

ing techniques have become customized from

smaller ad hoc, human-aided procedures to the

utilization of fully automated systems that are

able to convert entire websites into well-organized

data set. State-of-the-art web scraping tools are

not only capable of parsing markup languages or

JSON ﬁles but also integrating with computer

visual analytics (Butler 2007) and natural lan-

guage processing to simulate how human users

browse web content (Yi et al. 2003)

Web scraping, also known as web extraction or

harvesting, is a technique to extract data from the

World Wide Web (WWW) and save it to a ﬁle

system or database for later retrieval or analysis.

Commonly, web data is scrapped utilizing Hyper-

text Transfer Protocol (HTTP) or through a web

browser. This is accomplished either manually by

a user or automatically by a bot or web crawler.

Due to the fact that an enormous amount of het-

erogeneous data is constantly generated on the

WWW, web scraping is widely acknowledged as

an efﬁcient and powerful technique for collecting

big data (Mooney et al. 2015; Bar-Ilan 2001). To

adapt to a variety of scenarios, current web scrap-

ing techniques have become customized from

smaller ad hoc, human-aided procedures to the

utilization of fully automated systems that are

able to convert entire websites into well-organized

data set. State-of-the-art web scraping tools are

not only capable of parsing markup languages or

JSON ﬁles but also integrating with computer

visual analytics (Butler 2007) and natural lan-

guage processing to simulate how human users

browse web content (Yi et al. 2003)

Web scraping, also known as web extraction or

harvesting, is a technique to extract data from the

World Wide Web (WWW) and save it to a ﬁle

system or database for later retrieval or analysis.

Commonly, web data is scrapped utilizing Hyper-

text Transfer Protocol (HTTP) or through a web

browser. This is accomplished either manually by

a user or automatically by a bot or web crawler.

Due to the fact that an enormous amount of het-

erogeneous data is constantly generated on the

WWW, web scraping is widely acknowledged as

an efﬁcient and powerful technique for collecting

big data (Mooney et al. 2015; Bar-Ilan 2001). To

adapt to a variety of scenarios, current web scrap-

ing techniques have become customized from

smaller ad hoc, human-aided procedures to the

utilization of fully automated systems that are

able to convert entire websites into well-organized

data set. State-of-the-art web scraping tools are

not only capable of parsing markup languages or

JSON ﬁles but also integrating with computer

visual analytics (Butler 2007) and natural lan-

guage processing to simulate how human users

browse web content (Yi et al. 2003)

Web scraping, also known as web extraction or

harvesting, is a technique to extract data from the

World Wide Web (WWW) and save it to a ﬁle

system or database for later retrieval or analysis.

Commonly, web data is scrapped utilizing Hyper-

text Transfer Protocol (HTTP) or through a web

browser. This is accomplished either manually by

a user or automatically by a bot or web crawler.

Due to the fact that an enormous amount of het-

erogeneous data is constantly generated on the

WWW, web scraping is widely acknowledged as

an efﬁcient and powerful technique for collecting

big data (Mooney et al. 2015; Bar-Ilan 2001). To

adapt to a variety of scenarios, current web scrap-

ing techniques have become customized from

smaller ad hoc, human-aided procedures to the

utilization of fully automated systems that are

able to convert entire websites into well-organized

data set. State-of-the-art web scraping tools are

not only capable of parsing markup languages or

JSON ﬁles but also integrating with computer

visual analytics (Butler 2007) and natural lan-

guage processing to simulate how human users

browse web content (Yi et al. 2003)

**1.1 WEB SCRAPING**

Web scraping, also known as **web extraction** or harvesting, is a technique to extract data from the World Wide Web (WWW) and save it to a ﬁlesystem or database for later retrieval or analysis. Commonly web data is scrapped utilizing Hyper-text Transfer Protocol (HTTP) or through a web browser. This is accomplished either manually by a user or automatically by a web crawler. Due to the fact that an enormous amount of heterogeneous data is constantly generated on the WWW, web scraping is widely acknowledged as an efﬁcient and powerful technique for collecting big data.

To adapt to a variety of scenarios, current web scraping techniques have become customized from smaller ad hoc, human-aided procedures to the utilization of fully automated systems that are able to convert entire websites into well-organized data set. State-of-the-art web scraping tools are not only capable of parsing markup languages or JSON ﬁles but also integrating with computer visual analytics and natural language processing to simulate how human users browse web content.

Web scraping is a technique to fetch data from websites. While surfing on the web, many websites don’t allow the user to save data for personal use. One way is to manually copy-paste the data, which both tedious and time-consuming. Web Scraping is the automation of the data extraction process from websites. This event is done with the help of web scraping software known as web scrapers. They automatically load and extract data from the websites based on user requirements. These can be custom built to work for one site or can be configured to work with any website.

**1.1.1 What is Web Scraping?**

Whenever you want any information, you Google it and go to the webpage, which offers the most relevant answer to your query. You can view the data you needed, but what if you need to save it locally? What if you want to see the data of a hundred more pages?

Most of the webpages present on the internet don’t offer the option to save the data present there locally. To keep it that way, you’ll have to copy and paste everything manually, which is very tedious. Moreover, when you have to save the data of hundreds (sometimes, thousands) of webpages, this task can seem strenuous. You might end up spending days just copy-pasting bits from different websites.

**1.1.2 Why Web Scraping?**

* Web scraping collects and categorizes all the required data in one accessible location. Researching with a single, convenient location is much more feasible and more comfortable than searching for everything one-by-one.
* Web scraping simplifies the process of extracting data, speeds it up by automating it and creates easy access to the scrapped data by providing it in a CSV format. In simple terms, web scraping saves you the trouble of manually downloading or copying any data and automates the whole process.
* Automate data collection processes at scale
* Unlock web data sources that can add value to your business
* Make data-driven decisions
* Increasing reliance on analytics and automation are two big trends among businesses. Web scraping can enable both trends. Along with these reasons, web scraping has numerous applications that can affect all industries.

**1.2 TELEGRAM BOT:**

Bots are third-party applications that run inside Telegram. Users can interact with bots by sending them messages, commands and [**inline requests**](https://core.telegram.org/bots#inline-mode). You control your bots using HTTPS requests to **Telegram's Bot API.**

**1.2.1 WHAT CAN I DO WITH BOTS?**

To name just a few things, you could use bots to:

* **Get customized notifications and news**. A bot can act as a smart newspaper, sending you relevant content as soon as it's published.
* **Integrate with other services**. A bot can enrich Telegram chats with content from external services.  
  **Gmail Bot**, **GIF bot**, **IMDB bot**, **Wiki bot**, **Music bot**, **Youtube bot**, [**GitHubBot**](https://t.me/githubbot)
* **Accept payments from Telegram users**. A bot can offer paid services or work as a virtual storefront.    
  **Demo Shop Bot**, [**Demo Store**](https://t.me/teststore)
* **Create custom tools**. A bot may provide you with alerts, weather forecasts, translations, formatting or other services.  
  **Markdown bot**, **Sticker bot**, **Vote bot**, **Like bot**
* **Build single- and multiplayer games**. A bot can offer rich **HTML5 experiences,** from simple arcades and puzzles to 3D-shooters and real-time strategy games.  
  **GameBot**, **Gamee**
* **Build social services**. A bot could connect people looking for conversation partners based on common interests or proximity.
* **Do virtually anything else**. Except for dishes — bots are terrible at doing the dishes.

**ABSTRACT**

A dictionary is one of the most important tools during your time studying at a university. A good dictionary can help you understand your subject better, improve your communication and improve your grades by making sure you are using words correctly. Dictionary can help the learners in three major ways:

• They can help learners understand words that they meet in reading and listening.

• They can help learners find words that they need for speaking and writing.

• They can help learners remember words

Our Project idea is to scrape a dictionary website from Google and to extract the data from the website and feed through a telegram bot. The user enters a random word as an input in the Telegram bot page. The word which is given as input is processed and the meaning and correct pronunciation of the word is scraped from the dictionary website using python and web scraping modules and the extracted data is displayed as output to the user in the same telegram bot page. The website that is scraped is the famous and most popular dictionary website of the OXFORD COMMUNITY.

**CHAPTER 2**

**SYSTEM ANALYSIS**

**2.1 EXISTING SYSTEM**

In Existing system is the manual web data extraction process has two major problems. Firstly, it can’t measure costs efficiently and can escalate it very quickly. The data collection costs increase as more data is collected from each website. In order to conduct a manual extraction, businesses need to hire large number of staffs, this increases the cost of labour significantly. Secondly, each manual extraction is known to be error prone. Further, if any business process is very complex then cleaning up the data can get expensive and time consuming. The below figure explains the errors and data cleanup processes problems with the Manual method.

**In Europe**, Eurostat has supported the initiation of web scraping projects in the NSO’s of several EU Member States (Netherlands, Germany, Italy, Luxembourg, Norway, Sweden, Austria, Belgium, Finland and Slovenia). Of these countries, Germany, Italy and Netherlands and United Kingdom have circulated first results

**Germany and Italy** use a methodology that combine web scraping software (i Macros) with java programming to input, select, delete and store data within the price collection process. The Dutch have set up an own web crawling/robot framework using the software R.

**The British** are about to program own web scrapers using the software Python. The mentioned existing web scraping projects have in common that the development of data collection processes are out-sourced from the price index department to other units qualified to perform necessary programming and data managing tasks. Also, data validation, cleaning, editing and matching procedures are out-sourced as well as the new technology leads to quantitative data sets that cannot be handled any more using existing processes within the price index department.

**2.2 Literary Survey**

Web scrapping is the very useful technology in the field of getting the content from the different websites. The best features of the web scrapping technology are that can scrape the content which is required. Web scrapping is used by the many companies for business. One of the example of the web scrapping in the real estate listing gathering, It is a huge and growing web scrapping area. This is an area where the businesses are using web scraping to gather already listed. All the machine learning companies are using web scrapping to get the data. Email address gathering is another field of the application where once the emails are collected bulk emails are sent.

Website creators also uses the web scrapping where collecting data from the different social media websites, what is trending and what is in etc. Web scrapping is used in the one of the project in which it is used to scrape the content of particular category of book in the Amazon store. In another project web scrapping is used to scrape the contents from the Twitter on the basis of hash tag or by searching the keywords in the twitter. In the field of machine learning web scrapping is used in sentiment analysis field, where the data is scrapped from the websites.

Web scrapping is used in technologies such as Market research using web data in any of the industry. Even web scrapping technology is used in price comparing sites where it compares the price of item or room from different websites. In advance these applications use the web scraping to the scrape the content from the dependant websites. Various government and private watch dogs uses the web scrapping to monitor the malicious activities going on the internet.

**Netucon Company based at Ahmadabad** provides ultimate solutions to its customers and software development services with innovation and creativity. Netucon understands the requirements from the customers and clients and produces the software. They developed a LinkedIn connection creator this LCC is useful for scrapping CEOs, creating B2C contacts, Lead Generators, Digital marketers, Blogger who post their blogs on LinkedIn and so on.

**Renita Crystal Pereira** et. al., provided web scraping summary and techniques and tools that face several complexities as data extraction isn't that simple. These strategies guarantee that the data collected is correct, consistent and has better integrity, because there is a large amount of data present which is hard to handle and retain. Although there are a few problems faced by functional techniques that can be such as the elevated amount of web scraping be able to cause rigid harm to the websites. The measurement level of the web scraper will vary with the measurement units of the original source file, making it very difficult to interpret the data. Using social networking sites and internet is amplifying day by day like facebook, twitter, linked-in and some other, user knowledge is also high in the internet available from everywhere. This as well offers hackers an advantage in stealing information. Where the concept of rising income comes into being, social networking is important from a view of business point. Like with online shopping, it will also assist consumers in getting fast shopping and also save time. On the other hand, there is advantage in supporting the company and profiting from it.

**Kaushal Parikh** et. al., proposed a web scraping detection with the help of machine learning It is valuable for research dependent companies. Web scraping has forever been a difficult preventive attack. Every time a company places its data on internet, it is probable that it could be copied and pasted and then utilized in the other point of view without the corporation knowing itself about it. A lot of protection mechanisms have already been in place but some of them continue to be ignored. The significance of machine learning therefore steps in. Machine learning is quite effective on pattern detection. Therefore if we succeed in making the machine understand a cadence of intruder then it will avoid these types of threats from occurring. Web scraping solutions are aimed primarily at translating complex data obtained through networks into structured data that could be stored and examined in a central database. Web scraping solutions thus have a significant impact on the result of the cause.

**Sameer Padghan** et. al., projected an approach where data extraction is done from web pages in assistance with web scraping easily. This method would enable the data to be scrapped from numerous websites that will minimize human intervention, save time and also enhance the quality of data relevance. It will also support the user in gathering data from the site and to save the data to their intent and use it as the individual wishes. The scraped information may be used for database development or for research purposes and also for different similar activities. The scraping used would increase significantly and will often encroach on the framework to obtain the details. However the scraping can be stopped by using effective and safe-web scraping methods. This method should be treated as a blessing that must be used carefully for the advancement of human races.

**Anand Saurkar** et. al., discovered latest technique named Web Scraping. Web scraping is a quite important methodology used to produce structured data based on the unstructured data available on the internet. Scraping formed structured data, subsequently collected and evaluated in spreadsheets in central database. This research focuses on a summary of the data extraction process of web scraping, various web scraping strategies and most of the latest tools utilized to scrap web. The primary function of this methodology has been to get webbased information and integrate this into a specific repository. The authors addressed the basics of Web processing in this article. They concentrated on the Web scraping techniques. The final part of the paper presents a summary of the numerous technological resources that are available for effective web scraping in the industry.

**Federico Polidoro** et. al., concentrated on the outcomes of web scraping evaluation strategies with particular orientation to user electronics services and goods throughout the sector of commodity price studies. Although the research done has so far been performed in a small amount of time, that you can see in whatever followed, it has enabled to attain important, but not conclusive, novel efficiencies results. Web scraping strategies used in the growth analysis will provide exposure to a greater volume of data than that accessible in the existing data set, thus, with the potential to increase the growth estimate. This topic has been briefly addressed in the portions allocated to both Wutan Huatan Jisuan, of the examined items, but in reality interacting with this viewpoint requires a concern regarding the current survey architecture that does not require or only selectively permit the use of big data approaches within the existing sampling frameworks.

**Jan Kinne** et. al., Proposed a web extraction platform for the accurate and measurable mining of ecosystems for development. Researchers have put special emphasis on exploring a possible bias while examining technology structures across corporation website if all those types of companies could be measured using suggested method. Web extraction still has to deal with incredibly large and ultra-connected outer websites as a research tool, and the reality that limited broadband access continues to discourage companies from managing their internal websites and therefore preventing themselves from web mining research. The proposed system of research enables for an integrated, least expensive simulation of whole business communities, that could be conducted out more efficiently and in relatively short time periods compared to conventional techniques. This method is also conveniently extendable by checking the web pages of research institutions to model information communities. The key point in proposed system is to identify and extract certain bits of data from unstructured content on the site which exposes information regarding the current development practices of companies.

**Ingolf Boettcher** discovered that technique like web scraping can evolve. Web scraping innovation provides a range of choices and can satisfy various purposes: A web crawler's basic requirement is to automate the normally physical work of gathering price estimates and website article details. A web crawler's ultimate requirement will be to discover previously inaccessible pricing data outlets and include a census of all web-available price information. The actions to build web scraping for price analytics include significant analytical and administrative consequences. Any deployment of the approach involves a detailed preparation in various sectors. Elements of ethical and data protection need to be discussed first. Essential IT services and IT practice needed to manage the automated information gathering system must be calculated and must not be overlooked in the context of a research project.

**Erin Farley** et. al., destined to present web scraping to law enforcement researchers and illustrate what web scraping is about and how this technique works. Use of the web crawling by investigators in criminal justice is a fairly recent trend. Only a range of experiments wherein web scraping was used were identified in a literature review for criminal liberty related research using web scraping as an information gathering method. Although web scraping is usually seen as a method for collection of data to promote analysis and research, designing and implementing a web scraper includes technological abilities that researchers in the social sciences generally do not have. A strong level of expertise in computer science techniques like R or Python when developing source code is a necessity for creating a web scraper.

**2.3 MOTIVATION**

The Common and major motivation of Web scraping is a large amount of data is available on the web in a loosely structured form in HTML pages. While this data is largely meant for human consumption, some websites make it available in a structured format via “Web Services” using mechanisms such as REST and SOAP. These web services allow for programmatic interaction with the data.

However, a significant number of websites do not make such web services available, but the data in them are interesting nevertheless. In such cases, **Web Scrapers** are written to extract information in them and to load the data into more structured stores (ex. Databases) so something useful can be learned from it. The variations in the structure of HTML pages on the web disallow a generic one-size-fits-all algorithm from being used to extract the data; data must be extracted on a case-by-case basis. Web Scraping allows the retrieval of the contents of an HTML page, processing of the retrieved HTML for data extraction and persisting of extracted data to a flat-file (which can then be used to load the data into a more structured store such as an RDBMS).

Our motivation is to provide a complete all in one Bot API that provides the correct meaning and pronunciation of any number of words, for which the user requires.

**2.4 Proposed System**

Web Scraping (web harvesting or web data extraction) is a computer software technique to extract information from websites. Usually, such programming programs recreate human investigation of the World Wide Web by either executing low-level Hyper content Transfer Protocol (HTTP), or installing a completely fledged internet browser, like Internet Explorer or Mozilla Firefox. Web Scraping is firmly identified with web ordering, that lists data on the web utilizing about web crawler and is a widespread method received by most web indexes. Conversely, Web Scraping centers more around the change of unstructured information on the web, ordinarily in HTML design, into organized information that can be put away and investigated in a focal neighborhood data set or accounting page.

The proposed system for this project is **a web scraper that is able to access and extract data from websites using a web application as an interface for user interaction**. The extracted data is then displayed in the Telegram Bot API to the user as the website allows the data to be extracted and a python program is used to integrate both the scrapped network and the Telegram Bot API.

**2.4.1 The Page Ranking Algorithm**

PageRank is an algorithm which is being applied by Google for Searching for ranking web pages into search engine. It is the first algorithm that was applied by the google. Its name based on Larry Page, founders of Google. PageRank is a method which calculates significance for web pages. PageRank accepts that page has better rank if total of the rank of its out links is more. It is known the base for all present time Search Engines. The key approval for crucial webpages for received extra nodes from different web pages.

Google stated that PageRank works with the help of calculating the number and standard for nodes to web page to setup a dry idea for crucial web pages. Ranks pages depend upon number of out links indicating for them. The algorithm allocates pages a Total PageRank depends upon PageRank’s for outlines specify for page. The links for a page can be arranged in different types: Inbound links that is linked in between available site by external source page. Outbound links that can linked by the available one page to another page in identical web page and the links that has no outgoing link is known has dangling link. The Page rank of the web sites is evaluated as a total of the Page ranks for every page incoming node and split up by the number for outgoing node for every web pages.

**Page rank of P= (1-damping factor) + damping factor ∑ (page rank (Ki)/O(Ki))**

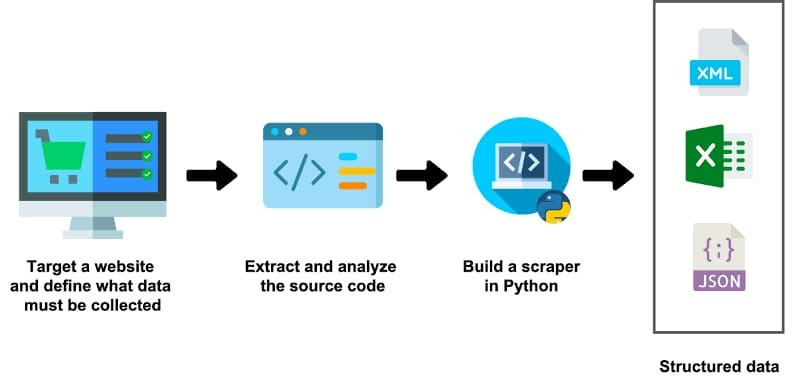
Where Page rank (P) is the PageRank of page P. Page rank (Ki) is the PageRank of pages Ki which link to page P O(Ki) is the number of outbound links on page Ki damping factor that may occur at intervals 0 and 1. It depends on the number of clicks, mostly at intervals to 0.85 n is the number of in links of page P.

Example: Page, A being referenced by pages B and C. C, B has 1,2 out links. Page rank value for page A is given as**: PR (A) =1-d + d (PR(C)/1 + PR (B)/2)**. The Algorithm will no rank entire web pages, however it’s negotiates to every page separately. And, PR (A) is recursively identify from Page rank of which pages connect to page A.

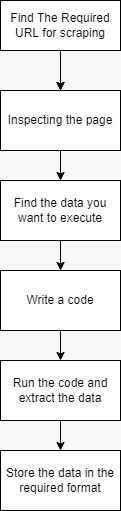
**2.5 SYSTEM ARCHITECTURE**

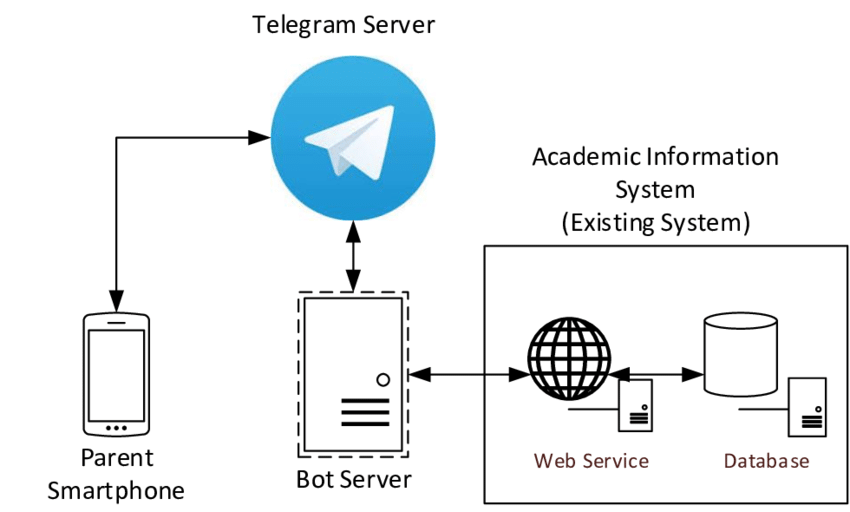
The architectural design of a system emphasizes the design of the [system architecture](https://en.wikipedia.org/wiki/System_architecture) that describes the [structure](https://en.wikipedia.org/wiki/Structure), [behaviour](https://en.wikipedia.org/wiki/Behavior) and more [views](https://en.wikipedia.org/wiki/View_model) of the proposed system and analysis.

**ARCHITECTURE DIAGRAM OF WEB SCRAPING**

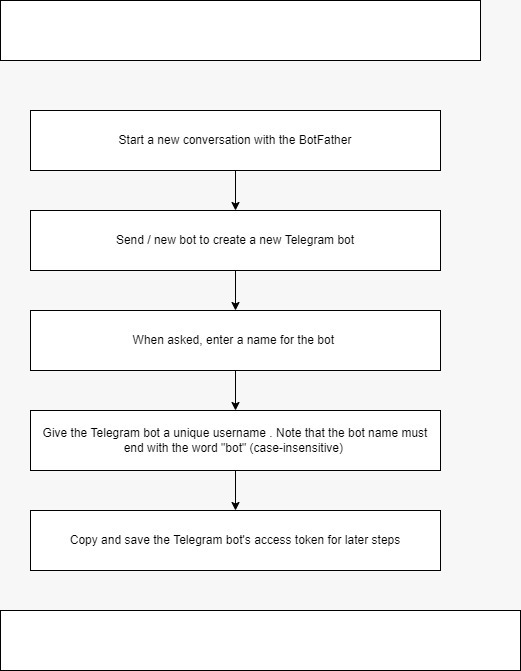


**METHODOLOGY CHART**

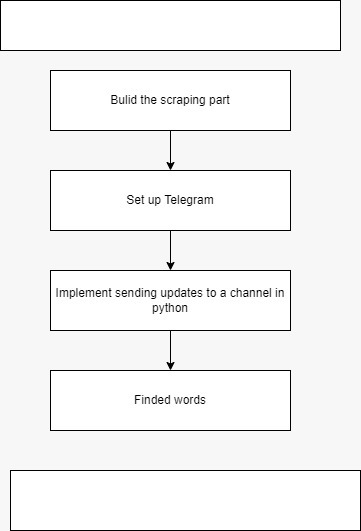


**ARCHITECTURE DIAGRAM OF TELEGRAM BOT**

**METHODOLOGY CHART**



**METHODOLOGY CHART**



**CHAPTER - 3**

**SYSTEM REQUIREMENTS**

A System Requirements specifies the project hardware and software requirements for developing the project. Supported operating system and run time environments where the project will run is on windows OS with python IDE. Since python IDE provides efficient way to implement AI And Machine Learning learning algorithms .

**3.1 HARDWARE REQUIREMENTS :**

RAM : 4 GB and above

Processor : 2.80 GHz and above

Hard Disk : 120 GB and above

CPU type : Intel I3 Processor or Intel I5 Processor .

Clock speed : 3.0 GH

Monitor type : 15 Inch color monitor

Keyboard type : Internet keyboard

• A PC that is connected to a network and can be used as a server to host the

application.

**3.2 SOFTWARE REQUIREMENTS**

Operating system : Windows 7, Windows 8, Windows 10 , Windows 11 .

Language : Python .

Documentation tool : Microsoft word 2007 and Above Versions .

Back end : Python.

Simulation Tool : Visual Studio Code and Chrome .

Software Needed : Visual Studio Code , Python ,Telegram bot API, .

**3.3** **FUNCTIONAL REQUIREMENTS**

• The system should prompt the user to log in before use for confidentiality, as the data is sensitive.

• The system should allow the user to enter a website to scrape.

• The system should crawl through and extract data from the desired website.

• The system should store data in a database with timestamps.

• The system should allow users to access and query the database from within the web application.

**3.4 NON-FUNCTIONAL REQUIREMENTS**

• The system should be user friendly and easy to navigate.

• The system should be easily maintained with minimal maintenance required.

• The system should be scalable and work efficiently under heavy workloads.

• The system should be easy to install and set up.

• The handling of user and scraped data should conform to the Data Protection

Act 1998.

**3.5 SOFTWARE FEATURES**

**3.5.1 PYTHON**

  Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don't need to declare the type of variable because it is a dynamic typed language.

### **EASY TO CODE**

Python is a very developer-friendly language which means that anyone and everyone can learn to code it in a couple of hours or days. As compared to other object-oriented programming languages like Java, C, C++, and C#, Python is one of the easiest to learn.

### **OPEN SOURCE AND FREE**

Python is an open-source programming language which means that anyone can create and contribute to its development. Python has an online forum where thousands of coders gather daily to improve this language further. Along with this Python is free to download and use in any operating system, be it Windows, Mac or Linux.

**3.5.2 Visual Studio Code:**

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE .Working with Python in Visual Studio Code, using the Microsoft Python extension, is simple, fun, and productive. The extension makes VS Code an excellent Python editor, and works on any operating system with a variety of Python interpreters .

**3.5.3 BEAUTIFUL SOUP :**

Beautiful Soup is a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) package for parsing [HTML](https://en.wikipedia.org/wiki/HTML) and [XML](https://en.wikipedia.org/wiki/XML) documents (including having malformed markup, i.e. non-closed tags, so named after [tag soup](https://en.wikipedia.org/wiki/Tag_soup)). It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for [web scraping](https://en.wikipedia.org/wiki/Web_scraping).

Beautiful Soup was started by Leonard Richardson, who continues to contribute to the project, and is additionally supported by Tidelift, a paid subscription to open-source maintenance.

Beautiful Soup is available on 64-bit [Linux](https://en.wikipedia.org/wiki/Linux), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Windows](https://en.wikipedia.org/wiki/Windows). Beautiful Soup 4 is published through PyPi, so if you can’t install it with the system packager, you can install it with easy install or pip. The package name is beautifulsoup4, and the same package works on Python 2 and Python 3. Make sure you use the right version of pip or easy install for your Python version (these may be named pip3 and easy\_install3 respectively if you’re using Python 3)

# 3.5.4 Telegram Bot API :

The Bot API is an HTTP-based interface created for developers keen on building bots for Telegram.

Bots are third-party applications that run inside Telegram. Users can interact with bots by sending them messages, commands and inline requests. You control your bots using HTTPS requests to Telegram's Bot API.

Telegram Bots are special accounts that do not require an additional phone number to set up. Users can interact with bots in two ways:

* Send messages and [commands](https://core.telegram.org/bots#commands) to bots by opening a chat with them or by adding them to groups.
* Send requests directly from the input field by typing the bot's @username and a query. This allows sending content from [inline bots](https://core.telegram.org/bots/inline) directly into any chat, group or channel.

Messages, commands and requests sent by users are passed to the software running on your servers. Our intermediary server handles all encryption and communication with the Telegram API for you. You communicate with this server via a simple HTTPS-interface that offers a simplified version of the Telegram API. We call that interface our [Bot API](https://core.telegram.org/bots/api).

Telegram bots are unique in many ways — we offer [two](https://core.telegram.org/bots#keyboards) [kinds](https://core.telegram.org/bots#inline-keyboards-and-on-the-fly-updating) of keyboards, additional interfaces for [default commands](https://core.telegram.org/bots#global-commands) and [deep linking](https://core.telegram.org/bots#deep-linking) as well as [text formatting](https://core.telegram.org/bots/api#formatting-options), [integrated payments](https://core.telegram.org/bots#payment-platform) and more.

**CHAPTER 4**

**SYSTEM DESIGN**

The term “Design” is defined as the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application.

**DATA FLOW DIAGRAM**

DFD are used to Specify Functions of the Information System and how data flow from function to function. A data flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart. The data flow diagram is part of the structured analysis modelling tools.

**DATA FLOW DIAGRAM AT THE INITIAL LEVEL (Level 0)**

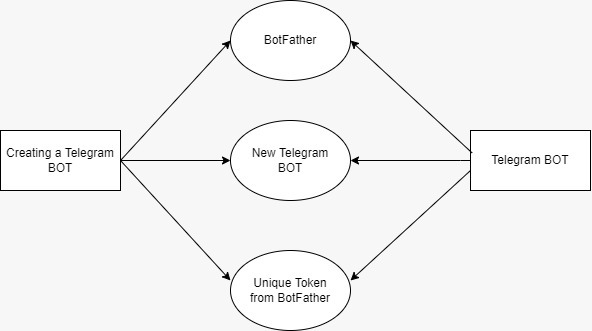


**Fig 4.1 Flow Diagram Level 0 For The Initial Level**

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities .

In this the diagram represents the whole system of [Web scraping](http://www.google.com) in a single level process.

**DATA FLOW DIAGRAM (LEVEL 1)**

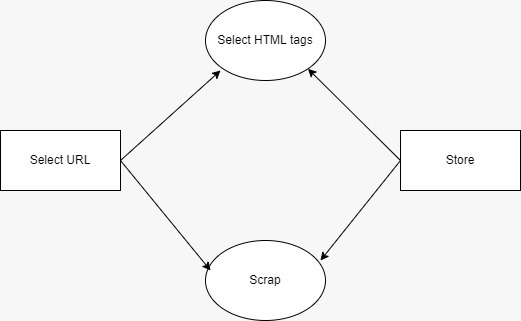


**Fig.4.2 Data Flow Diagram Level 1**

4.2 Data Flow Diagram Level 1 describes about the overall representation of each module and their functions. The level one data flow diagram has various modules and the respective results. As described previously, context diagrams (level 0 DFDs) are diagrams where the whole system is represented as a single process. A level 1 DFD notates each of the main sub-processes that together form the complete system

**DATA FLOW DIAGRAM (LEVEL 2)**

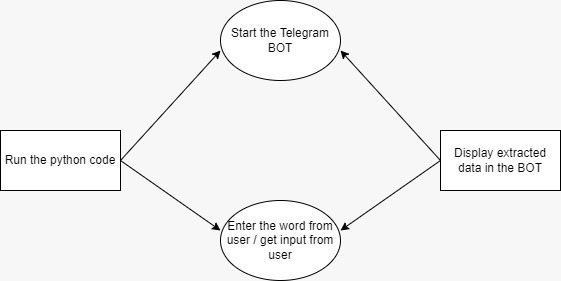
In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses. 2-level DFD: 2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning and We are going to see



**Fig.4.3 Data Flow Diagram Level 2**

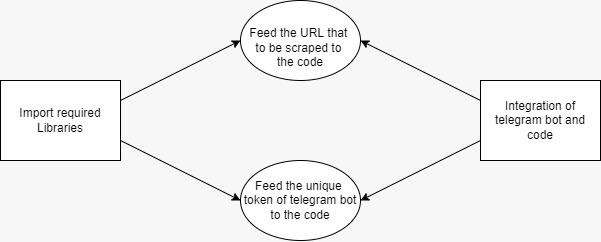
**DATA FLOW DIAGRAM (LEVEL 3)**

Data Flow Diagram Level 3 describes about the overall Transformers Actions of each module and their functions. The level one data flow diagram has various modules and the respective results. As described previously, context diagrams (level 0 DFDs) are diagrams where the whole system is represented as a single process. A level 1 DFD notates each of the main sub-processes that together form the complete system.



**Fig.4.4 Data Flow Diagram Level 3**

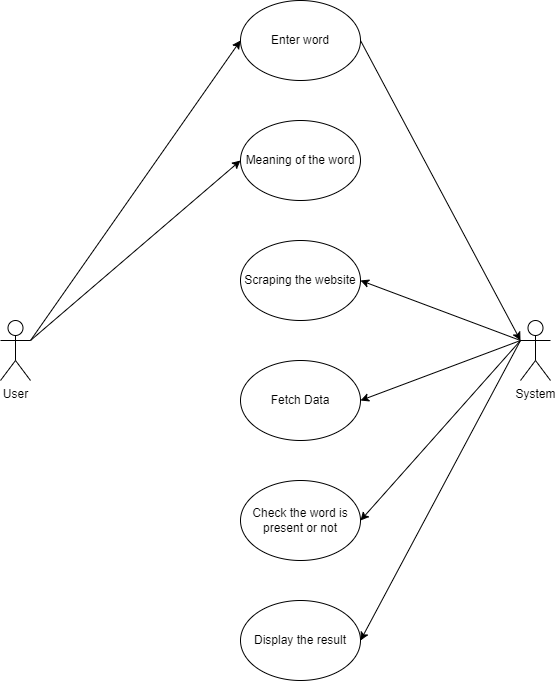
**DATA FLOW DIAGRAM (LEVEL 4)**



**Fig.4.5 Data Flow Diagram Level 4**

Data Flow Diagram Level 4 describes about the overall representation of each module and their functions. The level one data flow diagram has various modules and the respective results. As described previously.

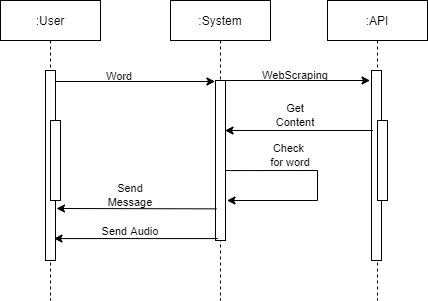
**4.6 USECASE DIAGRAM**

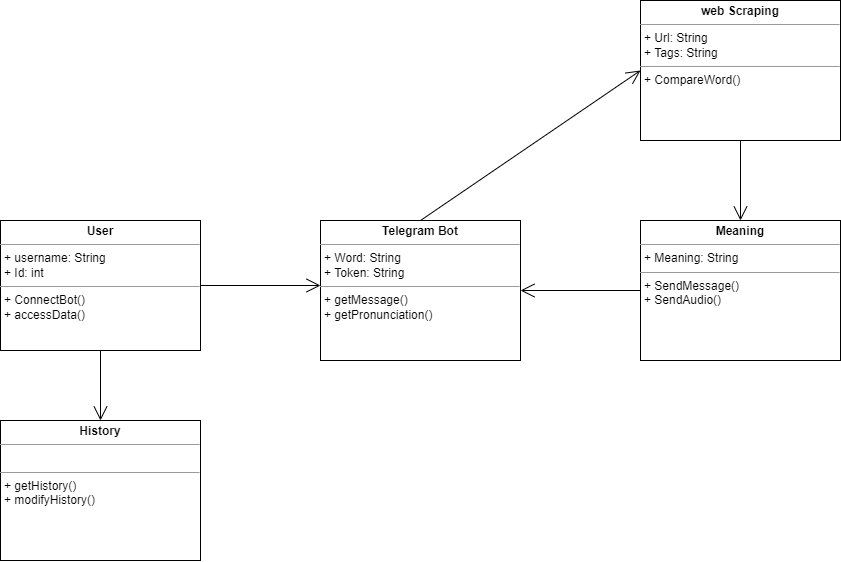


**Fig.4.6 Usecase Diagram**

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses .

* 1. **SEQUENCE DIAGRAM**

****

**4.8 CLASS DIAGRAM**

**CHAPTER-5**

**SYSTEM IMPLEMENTATION**

**LIST OF MODULES**

* Asyncio,
* Telepot.aio,
* Pprint,
* MessageLoop,
* Requests, And
* BeautifulSoup

**Asyncio**

Asyncio is a library to write concurrent code using the async/await syntax.

Asyncio is used as a foundation for multiple Python asynchronous frameworks that provide high-performance network and web-servers, database connection libraries, distributed task queues, etc.

Asyncio is often a perfect fit for IO-bound and high-level structured network code.

Asyncio provides a set of high-level APIs to:

* [run Python coroutines](https://docs.python.org/3/library/asyncio-task.html#coroutine) concurrently and have full control over their execution;
* perform [network IO and IPC](https://docs.python.org/3/library/asyncio-stream.html#asyncio-streams);
* control [subprocesses](https://docs.python.org/3/library/asyncio-subprocess.html#asyncio-subprocess);
* distribute tasks via [queues](https://docs.python.org/3/library/asyncio-queue.html#asyncio-queues);
* [synchronize](https://docs.python.org/3/library/asyncio-sync.html#asyncio-sync) concurrent code.

**Telepot.aio**

Telepot helps you build applications for Telegram Bot API. It works on Python 2.7 and Python 3. For Python 3.5+, it also has an async version based on asyncio. For a time, I tried to list the features here like many projects do.

**Pprint**

The [pprint](https://docs.python.org/3/library/pprint.html#module-pprint) module provides a capability to “pretty-print” arbitrary Python data structures in a form which can be used as input to the interpreter. If the formatted structures include objects which are not fundamental Python types, the representation may not be loadable. This may be the case if objects such as files, sockets or classes are included, as well as many other objects which are not representable as Python literals.

The formatted representation keeps objects on a single line if it can, and breaks them onto multiple lines if they don’t fit within the allowed width. Construct [PrettyPrinter](https://docs.python.org/3/library/pprint.html#pprint.PrettyPrinter) objects explicitly if you need to adjust the width constraint.

**MessageLoop**

The message loop is an obligatory section of [code](https://en.wikipedia.org/wiki/Source_code) in every [program](https://en.wikipedia.org/wiki/Computer_program) that uses a [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface) under [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows). Windows programs that have a [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) are [event-driven](https://en.wikipedia.org/wiki/Event-driven_programming). Windows maintains an individual [message queue](https://en.wikipedia.org/wiki/Message_queue) for each thread that has created a window. Usually only the first thread creates windows. Windows places [messages](https://en.wikipedia.org/wiki/Message_passing) into that queue whenever mouse activity occurs on that thread's window, whenever keyboard activity occurs while that window has focus, and at other times. A process can also add messages to its own queue. To accept user input, and for other reasons, each thread with a window must continuously retrieve messages from its queue, and act on them. A programmer makes the process do that by writing a [loop](https://en.wikipedia.org/wiki/Loop_(computing)) that calls GetMessage (which blocks for a message and retrieves it), and then calls DispatchMessage (which dispatches the message), and repeats indefinitely. This is the message loop. There usually is a message loop in the [main program](https://en.wikipedia.org/wiki/Main_program), which runs on the main [thread](https://en.wikipedia.org/wiki/Thread_(computing)), and additional message loop in each created [modal dialog](https://en.wikipedia.org/wiki/Modal_dialog). Messages for every window of the process pass through its message queue, and are handled by its message loop. A message loop is one kind of [event loop](https://en.wikipedia.org/wiki/Event_loop).

**Requests**

Requests allows you to send HTTP/1.1 requests extremely easily. There’s no need to manually add query strings to your URLs, or to form-encode your PUT & POST data — but nowadays, just use the json method!

Requests is one of the most downloaded Python packages today, pulling in around 30M downloads / week— according to GitHub, Requests is currently depended upon by 1,000,000+ repositories. You may certainly put your trust in this code.

**BeautifulSoup**

Beautiful Soup is a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) package for parsing [HTML](https://en.wikipedia.org/wiki/HTML) and [XML](https://en.wikipedia.org/wiki/XML) documents (including having malformed markup, i.e. non-closed tags, so named after [tag soup](https://en.wikipedia.org/wiki/Tag_soup)). It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for [web scraping](https://en.wikipedia.org/wiki/Web_scraping).

Beautiful Soup was started by Leonard Richardson, who continues to contribute to the project, and is additionally supported by Tidelift, a paid subscription to open-source maintenance.