# Read Me

This is a draft of a paper, periodically updated over time. It might not conform purely to the norms of an academic paper in the beginning but this work will be polished and edited numerous times during the summer 2021 semester and more if required and as such, does not reflect the final product.

# Abstract

The process of collecting data from the internet is known as web scraping. In a time when the internet is rich with data and there isn't enough time to look through it all, web scraping has become much more necessary and feasible to utilize in many applications. Beautiful Soup and the Python library requests are effective tools for the purpose. This project intends to extract text content from websites using web scraping to make it easier to detect a pattern in the information store or find relevant queries. The data is then stored into a database. As, User can get necessary information more easily and query more effectively. This project is focused on scraping Bengali web pages and attempting to determine the most efficient method of doing so.

# 1. Introduction

The amount of data kept online has only grown over time. The use of the internet and cloud servers has increased dramatically in recent years. As a result, searching for valuable data from available sources has become extremely challenging. Data is an essential part of any research, either it can be academic, marketing or scientific (SysNucleus, n.d.). People might want to collect and analyse data from multiple websites. The different websites which belongs to the specific category displays information in different formats. Even with a single website you may not be able to see all the data at once. The data may be spanned across multiple pages under various sections. Most websites do not allow to save a copy of the data, displayed in their web sites to your local storage (Penman et al., 2009). The only option is to manually copy and paste the data shown by the website to a local file in your computer. This is a very tedious job which can take lot of time. Web Scraping is the technique which people can extract data from multiple websites to a single spreadsheet or database so that it becomes easy to analyse or even visualize the data. (A Comparative Study on Web Scraping SCM de S Sirisuriya)

The aim of this study is to offers a review on web scraping techniques and software which can be used to extract data from web sites focusing the most on bangla websites and text contents.

# 2. Overview of Web Scraping

Web Scraping is a great technique of extracting unstructured data from the websites and transforming that data into structured data that can be stored and analysed in a database. Web Scraping is also known as web data extraction, web data scraping, web harvesting or screen scraping. Web scraping is a form of data mining. The overall goal of the web scraping process is to extract information from a websites and transform it into an understandable structure like spreadsheets, database or a comma-separated values (CSV) file

# 3. Related Works

**1. Visual Web Ripper Visual:** Web Ripper is one of the most advance web scraping software, created by Sequentum group in 2006 that provides functionality that allows you to scrape data from any websites like Business Directories, Simple Web Pages, Classified Sites, Forums and e-commerce site scraping (eBay, amazon, magento sites). Once data scraping finish, data can be exported to structured CSV, Excel, or XML format (List of Web Harvester, Data Scraper,Web Scraping Software and Tools)  
  
**2. Web Content Extractor**: Web Content Extractor (WCE) is a simple user-oriented application developed by Newprosoft. It has good wizard that guide user to setup scraper. You can scrape data from website with few clicks and Web Content Extractor is excellent for putting data into different formats like Excel, text, HTML formats, Microsoft Access database, Structured Query Language(SQL) Script File, MySQL Script File, Extensible Markup Language (XML) file, HTTP submit form and Open Database Connectivity (ODBC) Data source. (“List of Web Harvester, Data Scraper, Web Scraping Software and Tools,” n.d.) (“Software for Web Scraping,” n.d.).  
  
**3. Mozanda Web Scraper**: Mozanda Web Scraper is powerful web data extraction service. It can extract data from websites as well as PDFs. It has simple Point and selection interface so nontechnical can also make simple scrape. Mozenda runs your scraping project (agent) on their cloud environment which is the main difference of Mozanda from other scrapers. (“List of Web Harvester, Data Scraper, Web Scraping Software and Tools,” n.d.).   
  
**4. UiPath**: Robotic Process Automation UiPath can automatically log in to a web site, extract data spanning multiple webpages, filter and transform it into the format of user choice, before integrating it into another application or web service. UiPath resembles a real browser with a real user, so it can extract data that most automation tools cannot even see (Savinkin, n.d.). No programming is needed to create intelligent web agents using its drag-and-drop graphical designer-but the .NET hacker inside you has complete control over the data (“List of Web Harvester, Data Scraper,Web Scraping Software and Tools,” n.d.).   
  
**5. Out Wit Hub:** The OutWit Hub is a powerful Firefox extension Tool for everyone. The contents extracted from a web page are presented in an easy and visual way, without requiring any programming skills or advanced technical knowledge. Users can easily extract links, images, email addresses, data tables, etc. from series of pages without ever seeing the source code. Extracted data can be exported to CSV, HTML, Excel or SQL databases, while images and documents, are directly saved to your hard disk. The OutWit Hub is best to use for beginners in web scraping (“Software for Web Scraping,” n.d.).

# 4. System Design

To design a web scraper to serve our purposes, we will follow simple yet logical steps:  
  
Step 1: Define our data requirements

Step 2: Conduct a legal review

Step 3: Evaluate the technical feasibility

Step 4: Architect a solution & estimate resources

We will be trying to clarify and investigate the feasibility of this web scraping project you should always be answer questions like:

* What data do we require?
* From which websites would we like to obtain this data?
* How often would we like to extract this data? How do you want to consume the data?
* How will you verify that the extracted data is accurate? i.e. matches exactly the data on the target websites?
* How would you like to interact with the solution? i.e would you just like to receive data at a predefined frequency, or would you like to have control over the entire web scraping infrastructure and the associated source code?

**4.1 Approaches**We can take a few approaches towards building the type of web scraper we would require.   
  
**Mimicry Approach**: This category of scraper works thanks to predefined customised rules. The location of the data to be collected from a web page is preconfigured in the scraper. This mechanism is applied on DOM selectors which are deduced from click based leaning. This strategy is relatively efficient thanks to its neatness, but is less adapted when it comes to process multiple heterogeneous websites. Furthermore, if the source website modifies its graphic design, the engine should be reprogrammed to how to find the needed information. Tools such as Import.io or Mozenda use this approach.   
  
**Differential Approach**: This approach is based on the fact that two pages from the same website will only differ in content from the body of the page. According to this logic, the menu bars, the right or left columns, and the footers are supposed to be perfectly identical between two pages of the same website. The mechanism formerly consists in applying a masking algorithm that superimposes the two pages by removing only the differences.