

# UNDERSTANDING RELATIONSHIPS BETWEEN WEBSITES USING GRAPH PROCESSING

---

Umut Enes YEŞİL  
Murat Can ALAN  
Berfin NAZLI

Advisor: Bilgin AVENOĞLU

# Contents

1. Company Information
2. Problem
3. Analysis
4. Solution
5. Results and Conclusion
6. References
7. Demo

# Company Information

- IUG Network Mapping is a network mapping company dedicated to providing innovative solutions for visualizing and exploring web relationships. Our tools allow users to map and understand the connections between websites, IP addresses, URLs, and geographical locations.

## **Target Market:**

**Security Companies:** Offering services for website security analysis and threat detection.

**Digital Marketing Agencies:** Providing SEO optimization and online presence management services.

**Market Research Firms:** Conducting studies on online trends and competitor analysis.

**Government Agencies:** Involved in cybersecurity and monitoring online activities.

**Academic Institutions:** Conducting research on web structures and networks.

**Large Enterprises:** Interested in monitoring and analyzing their online presence and that of their competitors.

# Problem

- There are websites with databases and tools that can provide searching for IP's and URL's to get certain information about them such as country, city, time zone, domain name etc.
- But these services do not provide enough knowledge for marketing or SEO(Search Engine Optimization). With this project, we wanted to focus on bringing a lot of functionalities together and adding some unused features such as(Graph Visualizing) to provide a better picture for our users.
- We wanted to find a way to see a most linked website in a given url or ip groups.

# Analysis

- While there are some projects that uses graph visualization for their projects which some of them are similar to ours, we couldn't find any project that focuses to use it to develop a solution to our specific problem. Some of these projects are:

# Analysis

- **WebGraph Project:**

**Description:** The WebGraph project was developed by researchers at the University of Rome "La Sapienza" to study the structure of the World Wide Web.

**By Whom:** Researchers at the University of Rome "La Sapienza."

**When:** The project started in the early 2000s and has continued to evolve since then.

- **Wikipedia Clickstream Analysis:**

**Description:** Wikipedia clickstream analysis is a research effort to understand how users navigate Wikipedia.

**By Whom:** Researchers and data scientists interested in web user behavior.

**When:** The analysis has been ongoing for several years, with new studies and findings being published regularly.

- **Web Archive Analysis:**

**Description:** Web archive analysis involves studying archived web pages to understand historical web content and trends.

**By Whom:** Researchers, archivists, and historians interested in preserving and studying web history.

**When:** Web archive analysis has been conducted for many years, with ongoing efforts to improve techniques and tools.

- **Social Network Analysis:**

**Description:** Social network analysis studies relationships between individuals or organizations, including links to websites.

**By Whom:** Researchers in sociology, computer science, and related fields. Example: Facebook

**When:** Social network analysis has been a field of study for several decades, with ongoing research and applications in various domains.

# Analysis

- How is our project is different than them:
- **Visualization:** We specialize in visualizing website links, IP addresses, URLs, and location specific graphs, providing clear and intuitive graphs that highlight relationships.
- **Interactive Graph Exploration:** Our platform allows users to interact with the graphs, zooming in on specific areas, and exploring connections in depth.
- **Real-Time Data Updates:** Our system can update graphs in real time as new data is added, providing users with the most up-to-date information.(This is a case specific occasion that will be explained.)
- **More Than One Input Format:** We offer our users more than one input option to look for wider range of possibilities. Our users can create graphs by entering ip's, ip ranges, url's and city names.

# Solution

- Our program can take 4 types of inputs and do a get request, web scraping, and visualizing the linked websites. We can also execute our iplocationsearch code to search through ip ranges to find websites, scrapability, and location availability of those websites and group them in our database. That's how we provide a city search option in our website.

## Tools and Programming Languages that are used:

- Vite + React:

Vite: Provides a fast development environment with modern frontend tooling. It offers optimized configuration for React applications.

React: A popular JavaScript library used to create user interface (UI) components. React makes it easy to develop interactive and dynamic web applications with its component-based architecture and virtual DOM.

- HTML (HyperText Markup Language):

A standard markup language used to create the structure of web pages. HTML uses various elements (headings, paragraphs, links, images, etc.) to define the content structure.



# Solution

- JavaScript:

A programming language used to create dynamic and interactive web pages. It adds interactivity to web pages by running on the browser. It can also be used on the server side (e.g., with Node.js).

- Python Flask:

A minimalist Python web framework. It allows for fast and easy development of web applications and APIs. Flask is known for its flexibility and simplicity.

- Tailwind CSS:

A utility class-based CSS framework. Tailwind follows an approach that embeds styles rules into its components, making it easier to write CSS and allowing for less repetition and more modular code.

- MySQL:

A relational database management system (RDBMS). It is used to store, organize, and manage data in databases. Data operations are performed using the SQL (Structured Query Language) language.

# Solution

- JSX (JavaScript XML):

An extension with an HTML-like syntax used in React. JSX allows for writing HTML-like code inside JavaScript, which is then transformed into React components.

- Neo4j:

A graph database management system. It stores data in the form of nodes, relationships, and properties. It is ideal for modeling and querying complex relational data.

- Beautiful Soup:

A Python library used for extracting data from HTML and XML files. It is commonly used in web scraping operations and facilitates navigation and data extraction in parsed documents

# Results and Conclusions

## Project Results and Conclusions

- The project implements a comprehensive solution to visualize and analyze the relationships between URLs, IP addresses, and their connections using Neo4j as the graph database. Below are the key results achieved and conclusions drawn from the project:

- **Key Results:**

**Graph Visualization:** The application successfully visualizes the relationships between URLs and IPs as a graph. Each node represents a URL or IP address, and edges represent links between them. The web interface allows users to view these relationships interactively, showing both outgoing and incoming links.

**URL and IP Data Integration:** The system integrates data from multiple sources, including MySQL database for city-specific URLs and user-provided URLs or IPs. URLs are fetched from a MySQL database based on the specified city, enhancing the flexibility and scalability of data integration.

**Dynamic Depth Selection:** Users can dynamically select the depth of the graph, controlling how many levels of links are displayed. This allows for detailed or broad overviews of the URL connections.

# Results and Conclusions

**Outgoing and Incoming Links Analysis:** The application distinguishes between outgoing and incoming links. Users can query for nodes with the most outgoing or incoming links, providing insights into the most connected nodes in the network.

**IP Range Handling:** The system can handle and visualize links for a range of IP addresses, enabling comprehensive network analysis within specified IP ranges.

**Error Handling and Resilience:** The implementation includes robust error handling mechanisms for network requests, ensuring the system remains resilient against inaccessible URLs or IPs.

**MySQL Integration:** The project effectively reads URLs from a MySQL database based on city input, allowing for targeted analysis of specific geographic areas.

- **Key Features:**

**Add URL/City/IP:** Users can add URLs, cities, or IPs to the graph. The application fetches links from these inputs and visualizes the connections.

**Search by IP Range:** Users can input a range of IP addresses, and the application fetches and visualizes the links for each IP in the range.

**Clear Data:** Users can clear the current graph data, allowing for fresh analysis.

**Depth Control:** Users can set the depth of the graph to control how many levels of links are shown.

**View Link Counts:** The application provides a count of outgoing and incoming links for root URLs, helping identify the most influential nodes.

# Results and Conclusions

- **Conclusions:**

**Network Structure Insights:** The hierarchical and interconnected structure of the graph provides insights into the network's organization. Central nodes with high connectivity play crucial roles in the dissemination of information.

**Scalability and Flexibility:** The system is scalable and flexible, allowing users to add various types of data and adjust the depth of analysis. This makes it suitable for different use cases, from simple URL analysis to complex network studies. But the performance needs to be improved for large scale queries.

**Comprehensive Link Analysis:** By distinguishing between outgoing and incoming links, the application provides a comprehensive view of how different nodes interact within the network, revealing both sources and sinks of information.

**Error Handling:** The implemented error handling ensures that the system remains robust even when encountering inaccessible or invalid URLs/IPs, maintaining the integrity of the analysis.

- Overall, the project successfully achieves its goal of providing a robust, interactive tool for visualizing and analyzing URL and IP relationships within a network, leveraging the power of Neo4j for graph-based data representation.

# Results and Conclusions

## Advantages and Disadvantages of Our Project:

### Advantages:

**All in one:** Our project brings a lot of features from different tools and API services to one application.

**Specific Searching Possibility:** Our users can enter IP's, URL's, IP ranges and even city names to see the visualization of their desired group of websites.

**Graph Visualization:** Provides clear and intuitive visualizations of website links, IP addresses, URLs, and locations, making it easier for users to understand web networks.

# Results and Conclusions

## Disadvantages:

**Resource Intensive:** Handling large-scale web networks and real-time updates can be resource-intensive, potentially requiring significant computational power and robust infrastructure.

**Data Quality Dependency:** The effectiveness of visualizations depends heavily on the quality and accuracy of the input data. Poor data quality can lead to misleading or incorrect visualizations.

**Limited Scope of Data:** Due to the hardship that has been encountered while searching through ip ranges to find domain names, we must say that a lot of possible data could not be added to our database. While we think that if we had a bigger team with more facilities this problem wouldn't be an issue, the project as it is has a very limited scope of preprocessed data.

# Future For The Project

## **Incorporate Data Analysis:**

- Adding built-in data analysis tools to provide insights directly from the visualized data, such as identifying key nodes, clusters, and patterns in the web network.

## **Machine Learning Integration:**

- Implementing machine learning algorithms to predict trends, detect anomalies, and automate link classification, enhancing the analytical capabilities of the platform.

## **Enhanced Data Sources:**

- Integrating additional data sources, such as social media links, API data, or user behavior analytics, to provide a more comprehensive view of web networks.

## **Performance Optimization:**

- Optimizing the platform for better performance, allowing it to handle even larger datasets and more complex networks without compromising on speed or responsiveness.



# References

- **WebGraph Project:**

Boldi, Paolo, et al. "The webgraph framework I: Compression techniques." Proceedings of the 13th international conference on World Wide Web. 2004.

Boldi, Paolo, et al. "The webgraph framework II: Parallelization." Proceedings of the 13th international conference on World Wide Web. 2004.

- **Wikipedia Clickstream Analysis:**

West, Robert, et al. "Human wayfinding in information networks: A case study of Wikipedia." Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. 2015.

Priedhorsky, Reid, et al. "Inferring web browsing activity from usage patterns." Proceedings of the 2007 ACM conference on Electronic Commerce. 2007.

- **Web Archive Analysis:**

Klein, Martin, et al. "The web in the past and the present: A quantitative study of the World Wide Web." Computer Networks 50.18 (2006): 3605-3624.

Masanés, Julian, et al. "A quantitative approach to web archiving." International Journal on Digital Libraries 7.3 (2007): 195-207.

- **Social Network Analysis:**

Wasserman, Stanley, and Katherine Faust. Social network analysis: Methods and applications. Cambridge university press, 1994.


Newman, Mark. Networks: An Introduction. Oxford University Press, 2010.

# References

- **Vite:** Vite Documentation - vitejs.dev
- **React:** React Documentation - reactjs.org
- **JavaScript:** JavaScript Guide on MDN - <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
- **Flask:** Flask Documentation - flask.palletsprojects.com
- **Tailwind CSS:** Tailwind CSS Documentation - tailwindcss.com
- **MySQL:** MySQL Documentation - dev.mysql.com/doc
- **Neo4j:** Neo4j Documentation - neo4j.com/docs
- **Beautiful Soup:** Beautiful Soup Documentation - [www.crummy.com/software/BeautifulSoup/bs4/doc](http://www.crummy.com/software/BeautifulSoup/bs4/doc)
- **JSX:** JSX Documentation - reactjs.org/docs/introducing-jsx.html

# Demo

IP URL GRAPH



Select depth:  
1

Enter link or IP here

Add City Add Url Add Ip

Get Graph

Outgoing Link

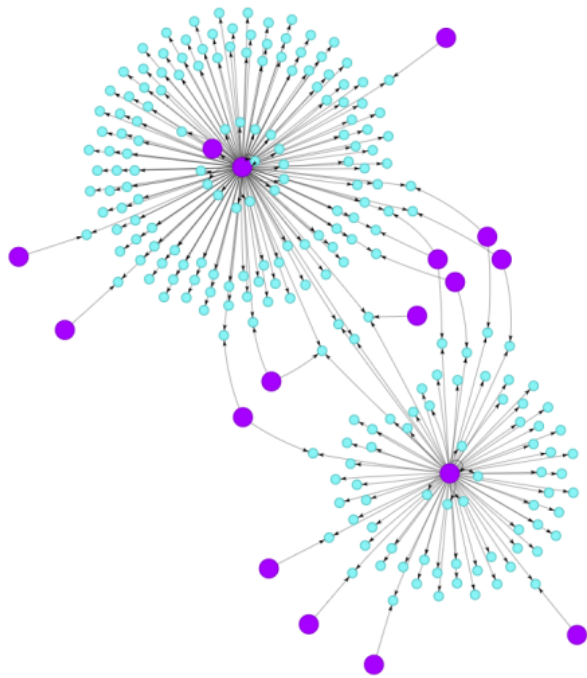
Incoming Link

Delete Data

Enter start IP here Enter end IP here

Search IP Range

## Graph Visualization



## Links

Home Tutorial Graph Viewer

- LINK: hurriyet.com.tr  
Has Link Count: 147
- LINK: milliyet.com.tr  
Has Link Count: 75
- LINK: youtube.com  
Has Link Count: 3
- LINK: facebook.com  
Has Link Count: 2
- LINK: instagram.com  
Has Link Count: 2