SCTR's Pune Institute of Computer Technology, Dhankawadi, Pune

AN INTERNSHIP REPORT ON

Inhouse Project

SUBMITTED BY

Name: Ojasvi Deshpande

Class: TE 02

Roll no: 31221

Under the guidance of

Dr. K. C. Waghmare

DEPARTMENT OF COMPUTER ENGINEERING

ACADEMIC YEAR 2022-23

DEPARTMENT OF COMPUTER ENGINEERING

SCTR's Pune Institute of Computer Technology
Dhankawadi, Pune
Maharashtra 411043

CERTIFICATE

This is to certify that the SPPU Curriculum-based internship report entitled

"Inhouse Project"

Submitted by
Ojasvi Deshpande
Roll no: 31221

has satisfactorily completed the curriculum-based internship under the guidanceof Dr. K. C. Waghmare towards the partial fulfillment of third year Information Technology Semester VI, Academic Year 2022-23 of Savitribai Phule Pune University.

Dr. K. C. Waghmare	**Dr. G. V. Kale**
Internship Guide	Head
PICT, Pune	Department of Computer Engineering, PICT, Pune

Place: PICT, Pune
Date: 18/05/2023

ACKNOWLEDGEMENT

It gives me great pleasure in presenting the internship report on Inhouse Project. First of all I would like to take this opportunity to thank my internship guide Dr. K.C. Waghmare for giving me all the help and guidance needed. I am really grateful for his kind support and valuable suggestions that proved to be beneficial in the overall completion of this internship. I am thankful to our Head of Computer Engineering Department, Dr. G.V.Kale, for her indispensable support and suggestions throughout the internship work. I would also genuinely like to express my gratitude to the Department Internship Coordinator, Prof. P. P. Joshi, for her constant guidance and support and for the timely resolution of the doubts related to the internship process. Finally, I would like to thank my mentor, Prof. D.D. Kadam for her constant help and support during the overall internship process.

Ojasvi Deshpande
31221

Contents

1. Title

Web application to get Hindex and i10 index of authors' publications from google scholar.

2. Introduction

The web application developed aims to provide the h-index and i10 index of authors' publications by retrieving data from Google Scholar. The application will have a user interface where users can enter either the name of the author, or the hindex based on which authors are to be displayed or i10 index . It will then search for the parameters in the database and if not found, it will scrape the data from the platform, fetch the required information and store it into the database if available.

Traditional ways of finding out the authors of PICT on google scholars included searching each author individually which was quite time consuming. As the no of citations and indices keeps on changing constantly and is huge in amount, the data also needed to be reevaluated constantly. Due to this, a web application that would provide all this information correctly and quickly was needed.

To accomplish this, we set up a web server using NodeJs language and its libraries. The server will handle HTTP requests and responses for the web application. Users will interact with the application through a user interface, where they can input required information parameters and get the result page about the authors' information along with a report pdf. The application will integrate with Google Scholar by querying the user requirements. The retrieved publication data, including citation counts for each paper, will be used to calculate the h-index and i10 index. The results will be displayed to the user along with other filters.

To ensure a smooth user experience, error handling mechanisms will be implemented to address cases where the author's profile is not found.Regular maintenance and updates would be performed to accommodate any changes in the website structures of Google Scholar.

3. Problem Statement

Develop a web application to get hindex and I index of authors' publications from google scholar.

4. Objectives and Scope

4.1 Objectives

- **Create a time and effort efficient website :** Develop a web application that would accurately and quickly deliver authors' information. Traditional methods of locating the authors of PICT on Google Scholar involved time-consuming individual author searches as the data required to be updated often due to large number of citations and indexes which are constantly changing.
- **Design the user interface:** Create a web page with a form where users can enter the author's name or hindex or i10index to retrieve the authors' metrics.
- **Handle form submission:** Implement NodeJs code to handle form submission. Capture the input from the form and initiate the process of retrieving publication metrics from Google Scholar.
- **Data management in the database :** Look up for the required information in the database initially and then if it is unavailable, search for it on Google Scholar site and scrape it.
- **Scrape Google Scholar data:** Use a library like Puppeteer to programmatically control a headless Chrome browser and scrape the necessary data from the Google Scholar website. With Puppeteer, you can search for the author's profile, extract the required information, and navigate through the pages to obtain publication details.
- **Get the results on website along with a report pdf:** Display the results based on information parameters (author's name, hindex, i10index) provided on the web page. Provide an option to the user to get a pdf of the report.
- **Keeping the information updated :** As these information keep changing constantly, it is necessary to keep the information in the database constantly updated.

4.2 Scope

The scope of this project has been to understand the requirement of the web application and the idea behind it. Initially, searching the authors of PICT on Google Scholar involved time-consuming individual author searches as the data required to be updated often due to large number of citations and indexes which are constantly changing. The aim of the project was to develop a website that can retrieve the h-index and i10 index of authors' publications from Google Scholar saving a lot of time and efforts. The application will have a user interface where users can input the author's name of index values, and the application will retrieve the relevant information from the database or by scraping data from the website. The project will involve setting up a web server and integrating with the Google Scholar. The application will be programmed using NodeJs programming language and relevant libraries such as puppeteer.

The project's further objectives includes grouping the information according to citation counts and other domains. Further, it also involves executing different data analysis techniques on the data gathered in the database and using the results to make appropriate inferences.

5. Methodological Details

5.1 Designing and Developing Website :

Steps to start a Node Project :

- Set up Node.js: Ensure that you have Node.js installed on your system. You can download the latest version from the official Node.js website (https://nodejs.org). Follow the installation instructions for your operating system.

- Create a project directory: Choose a directory where you want to create your Node.js project. Open a terminal or command prompt, navigate to the desired directory, and create a new folder for your project.
- Initialize a new Node.js project: Use the npm init command to initialize a new Node.js project in your project directory. This command will guide you through a series of prompts to set up your project's details, such as package name, version, entry point, dependencies, etc.
- Install dependencies: If your project requires any third-party dependencies or libraries, you can use the npm install command to install them. For example, if you want to install Express.js as a dependency, you would run: npm install express.
- This command will download and install Express.js and its dependencies in a folder named node_modules within your project directory.
- Create your application code: Start writing your Node.js application code. The entry point of your application is typically a JavaScript file, commonly named index.js or app.js. You can create this file in your project directory and start coding your Node.js application logic.
- Run your Node.js application: To run your Node.js application, execute the main JavaScript file using the node command.

About puppeteer :

Puppeteer is a Node.js library developed by the Chrome team at Google. It provides a high-level API for automating and controlling headless versions of the Chrome or Chromium browsers. With Puppeteer, you can programmatically interact with web pages, perform various tasks such as generating screenshots and PDFs, scraping data, running automated tests, and more.

Steps to use puppeteer :

- Install Puppeteer: Install the Puppeteer library as a dependency for your project using the npm install command. Run the following command in your project directory. This command will download and install Puppeteer and its dependencies in the node_modules folder of your project.

- Import Puppeteer: In your JavaScript file, import Puppeteer using the require or import statement.
- Use Puppeteer: With Puppeteer imported, you can now use its API to control and interact with headless Chrome or Chromium. Here's a simple example that launches a browser, opens a new page, navigates to a website, and takes a screenshot:

  ```
          const puppeteer = require('puppeteer');
          async function run() {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://example.com');
  await page.screenshot({ path: 'screenshot.png' });
  await browser.close();
  }
  run();
  ```

  In this example, puppeteer.launch() launches a headless browser instance, browser.newPage() creates a new page, page.goto() navigates to the specified URL, and page.screenshot() captures a screenshot of the page. You can explore Puppeteer's API documentation (https://pptr.dev/) to discover more features and methods for web automation, page manipulation, network interception, and other tasks. Run the Node.js script.
- By following these steps, you can start using the Puppeteer library in your Node.js project to automate browser tasks, scrape data, generate screenshots or PDFs, and perform various other web-related operations.

6. Modern engineering tools used

1. HTML:

HyperText Markup Language, or HTML. With the aid of a markup language, it is used to design web pages. Markup lan- guage and hypertext are combined to create HTML. The link between web pages is defined by hypertext. The text docu- ment inside the tag that specifies the structure of web pagesis defined using a markup language. This language is used to annotate (add notes to) material so that a computer can com- prehend it and modify the content as necessary. Most markup languages, like HTML, can be read by people. The language employs tags to specify what text processing is required.

2. CSS :

CSS, or cascading style sheets, is an acronym. It is a languagefor creating style sheets that describe the layout and appear- ance of markup-language documents. It gives HTML an addi- tional feature. Typically, it works with HTML to modify the look and feel of online pages and user interfaces. Any XML document type, including plain XML, SVG, and XUL, can be used with it.To construct user interfaces for web apps and manymobile applications, most websites combine CSS, HTML, andJavaScript.

3. JavaScript:

JavaScript, commonly referred to as the scripting language for websites, is a simple, single-threaded, interpreted, and cross- platform computer language. Many non-browser environments also use it, and it is widely accepted for web page develop- ment. JavaScript is a dynamically typed, weakly typed lan- guage. Both client-side and server-side development can be done with JavaScript. Both imperative and declarative lan- guages can be used with JavaScript.

4. NodeJS:

Node.js is an open-source, server-side runtime environment that allows developers to build and run JavaScript applications outside of a web browser. It uses the V8 JavaScript engine, developed by Google, which compiles JavaScript code into machine code, making it highly efficient.

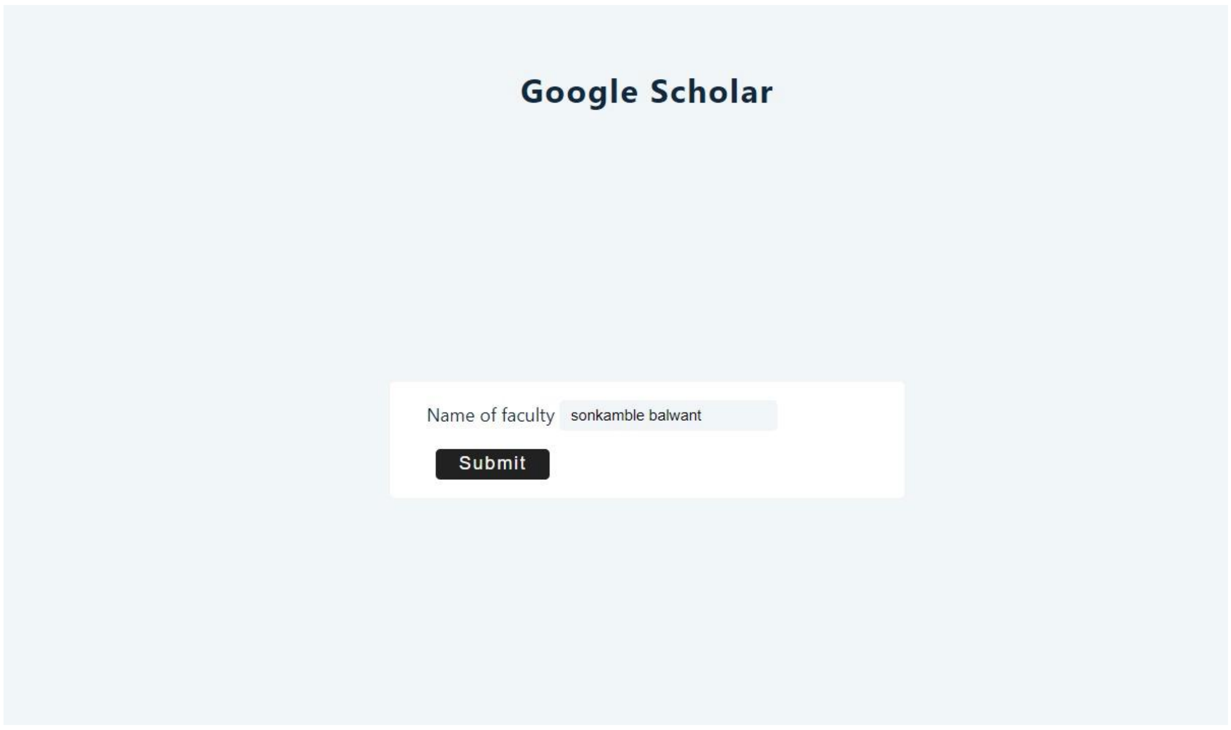
Here are some key features and concepts associated with Node.js:

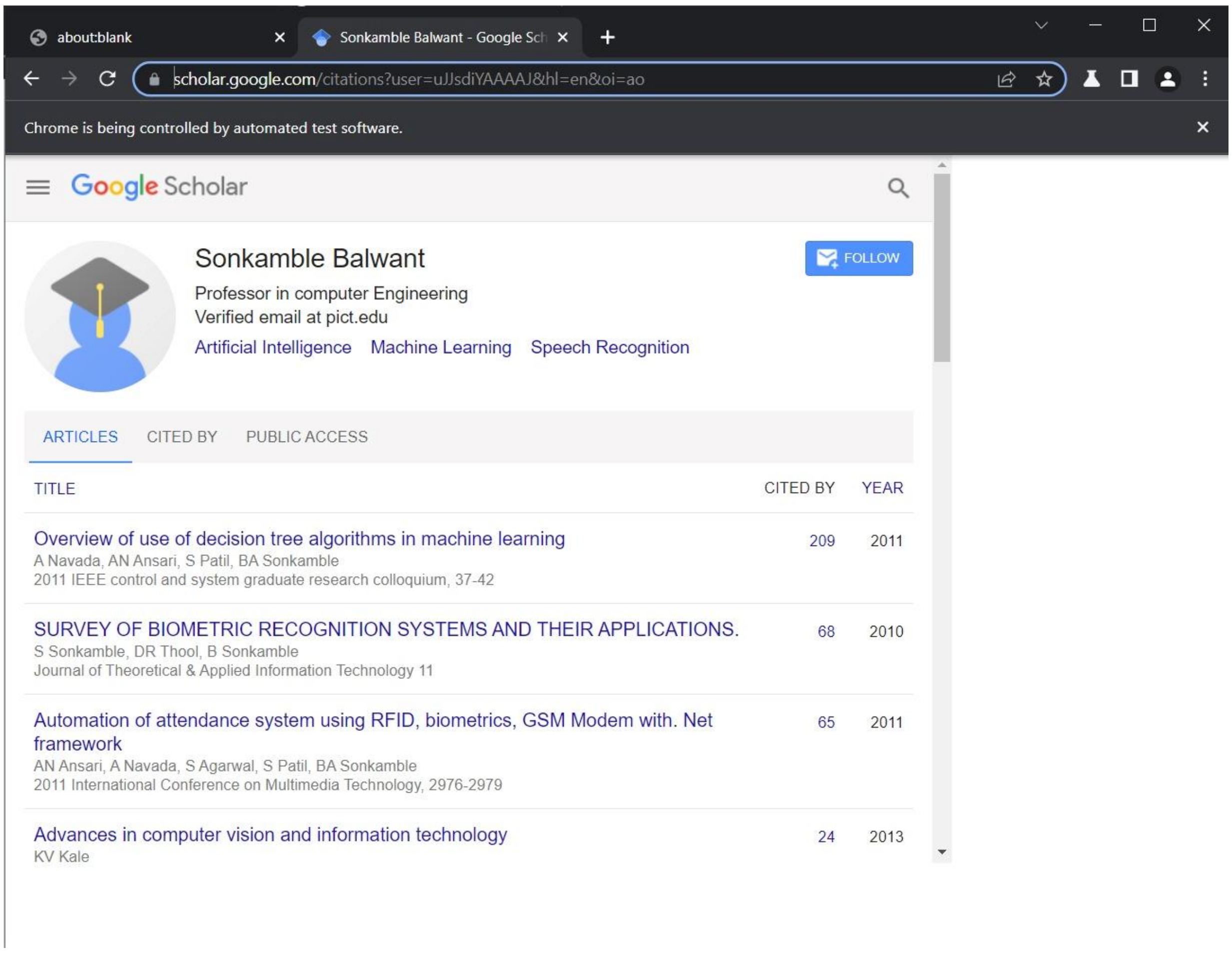
i. **Asynchronous programming:** Node.js is known for its ability to handle a large number of concurrent connections efficiently by using an event-driven, non-blocking I/O model. This means that instead of waiting for one operation to complete before moving to the next, Node.js can initiate multiple operations and process them in parallel, resulting in high performance and scalability.

ii. **NPM (Node Package Manager):** NPM is the default package manager for Node.js, allowing developers to easily manage and install third-party libraries and modules needed for their projects. It provides access to a vast ecosystem of reusable modules, making it easier to build applications by leveraging existing code.

iii. **Cross-platform Compatibility:** Node.js is available on multiple platforms, including Windows, macOS, and various Linux distributions. This cross-platform compatibility allows developers to write applications on one platform and easily deploy them on different environments.

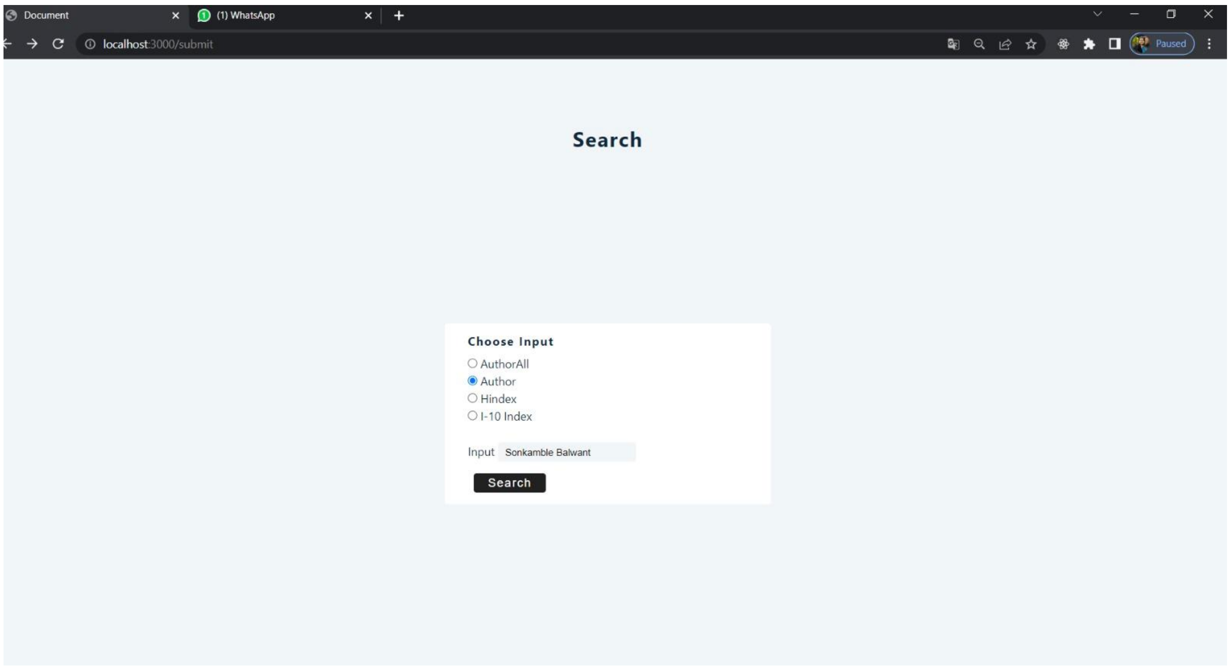
7. Outcome/ results of internship work (screen-shots of work done)




about:blank
Sonkamble Balwant - Google Sch
scholar.google.com/citations?user=uJJsdiYAAAAJ&hl=en&oi=ao
Chrome is being controlled by automated test software.
Google Scholar
FOLLOW
ARTICLES
CITED BY
PUBLIC ACCESS

Sonkamble Balwant
Professor in computer Engineering
Verified email at pict.edu
Artificial Intelligence Machine Learning Speech Recognition

TITLE	CITED BY	YEAR
Overview of use of decision tree algorithms in machine learning A Navada, AN Ansari, S Patil, BA Sonkamble 2011 IEEE control and system graduate research colloquium, 37-42	209	2011
SURVEY OF BIOMETRIC RECOGNITION SYSTEMS AND THEIR APPLICATIONS. S Sonkamble, DR Thool, B Sonkamble Journal of Theoretical & Applied Information Technology 11	68	2010
Automation of attendance system using RFID, biometrics, GSM Modem with. Net framework AN Ansari, A Navada, S Agarwal, S Patil, BA Sonkamble 2011 International Conference on Multimedia Technology, 2976-2979	65	2011
Advances in computer vision and information technology KV Kale	24	2013

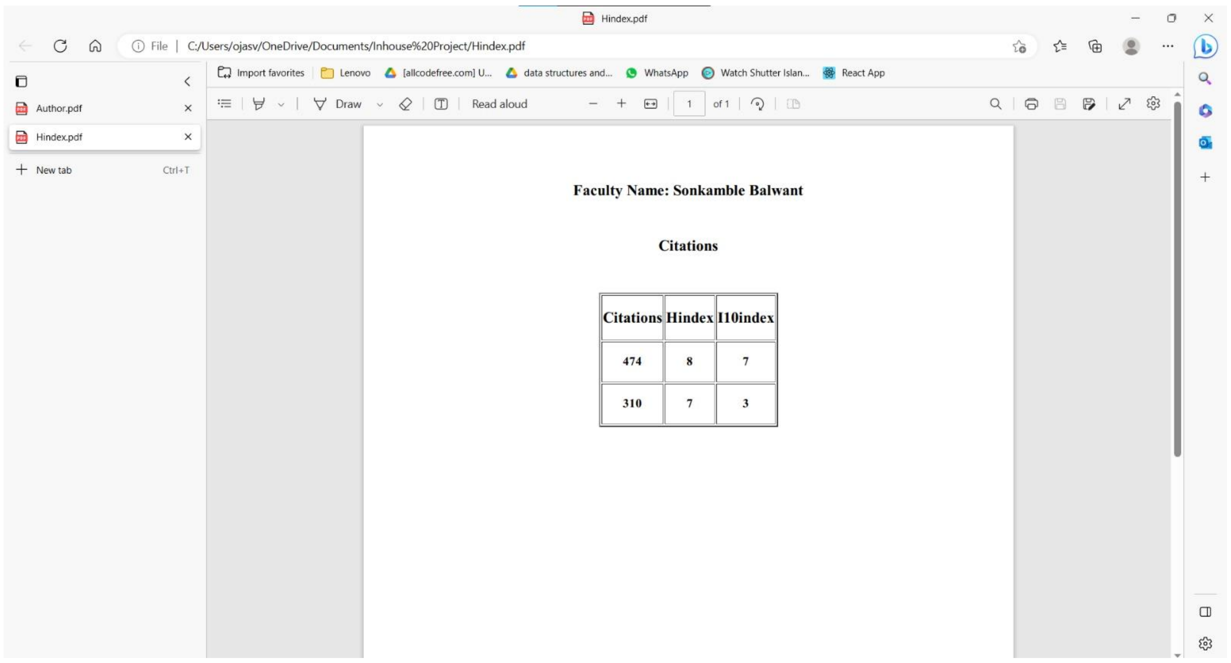


Search

Choose Input

AuthorAll [✓ Author] Hindex I-10 Index

Input: Sonkamble Balwant


Hindex.pdf
C:/Users/ojasv/OneDrive/Documents/Inhouse%20Project/Hindex.pdf
Author.pdf
Hindex.pdf
New tab
Draw
Read aloud

Faculty Name: Sonkamble Balwant

Citations

Citations	Hindex	I10index
474	8	7
310	7	3

8. Conclusion

In conclusion, the project aims to develop a comprehensive web application that will greatly enhance the process of retrieving and evaluating research output from authors by providing their h-index and i10 index. By integrating with Google Scholar, the application will enable users to quickly and accurately assess the impact and productivity of authors' publications.

The web application will feature a user-friendly interface where users can input the name of the author or specific index values to retrieve relevant information. This eliminates the need for manual and time-consuming searches for each author, streamlining the research evaluation process. Additionally, the application will store the retrieved data in a database, reducing the need for repeated scraping and ensuring efficiency in subsequent searches.

To implement the project, a web server will be set up using NodeJs and its libraries, enabling the handling of HTTP requests and responses. Through seamless integration with Google Scholar, the application will query the necessary parameters to fetch publication data, including citation counts for each paper. These data will then be utilized to calculate the h-index and i10 index, providing valuable metrics for researchers, academic institutions, and funding agencies.

Regular maintenance and updates will be performed to adapt to any changes in the website structures of Google Scholar, ensuring the continued functionality and accuracy of the application.

Overall, the project addresses the limitations of traditional methods for evaluating research output, such as the time-consuming nature of individual searches and the constant need for reevaluation due to changing citation counts. By leveraging the power of web scraping, database management, and data analysis techniques, the application significantly enhances the efficiency and accuracy of assessing authors' impact and productivity. It serves as a valuable tool for researchers, institutions, and funding agencies in making informed decisions and promoting scientific advancement.

9. References

1. http://www.w3schools.com
2. https://www.javatpoint.com/javascript-tutorial
3. http://www.w3.org/standards/semanticweb/
4. http://www.youtube.com
5. http://www.scholar.google.com
6. https://drive.google.com/drive/u/0/folders/1POKm4t71uTgsnd9PYo8eD9RsOkwvL4_T
7. https://nodejs.dev/en/learn/
8. https://npmjs.com