Zeeshan Malik
01-235182-080
Nouman Malik
01-235182-056

Data Annotation System for Schema.org Based SEO

Bachelor of Science in Information Technology

Supervisor: Burhan Ud Din Abbasi

Department of Computer Science
Bahria University, Islamabad

29, April 2022

Certificate

We accept the work contained in the report titled “Data Annotation System for Schema.org Based SEO ”, written by Mr. Zeeshan Malik AND Mr. Nouman Malik as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Information Technology.

Approved by . . . :

Supervisor: Burhan Ud Din Abbasi (Lecturer)

Internal Examiner: Ms. Maryam Bibi (Lecturer)

Internal Examiner: Ms. Sara Mehmood (Lecturer)

External Examiner: Name of the External Examiner (Title)

Project Coordinator: Miss Zubariya Inayat (Sr. Lecturer)

Head of the Department: Dr. Arif ur Rahman (Sr. Associate Professor)

Abstract

A complete list of values and items that can be mark up with schema tags is at scheme.org. Schema.org is a way to feature machine-readable facts to web pages that is processed by the main search engines like Google to enhance or boost search performance. It is used to create, maintain and promote schema for structured data on the internet, on web pages, in email messages. In order to facilitate this process schema markups are used which help search engines to grasp content and represent it in a better way within the search results.

Schema markup creates a greater specific description in page. When there is a use of schema markup, Google can simply perceive the web page's content. This data annotation system we deployed will help the user to get the data available on site by labeling it in multiple formats.

Acknowledgments

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to our final year project supervisor Mr Burhan Ud Din Abbasi, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project especially in writing this report. Furthermore, a special thanks goes to my teammate Zeeshan Malik who help me to assemble the parts and gave suggestion about tasks. I have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advice.

Nouman Malik
Islamabad, Pakistan

29, April 2022

*“We think someone else, someone smarter than us,
*

someone more capable, someone with more resources will solve that problem.
But there isn’t anyone else.”

Regina Dugan

Contents

Abstract **i**

1 Introduction **1**

1.1 Project Overview 1

1.2 Problem Description 3

1.3 Project Objectives 4

1.4 Project Scope 5

1.5 Methodology 5

2 Literature Review **6**

2.1 Schema.org 7

2.1.1 How it's Organized: 7

2.2 Schema Validator 10

2.3 Data Annotation System 10

3 Requirement Specifications **12**

3.1 Requirement Specifications 13

3.1.1 Functional Requirements: 13

3.1.2 Non-Functional Requirements: 13

4 Design **15**

4.1 System Architecture 15

4.1.1 System Context Diagram 16

4.1.2 Labeling Procedure 17

4.2 Design Constraints 18

4.3 Design Methodology 18

4.4 GUI Design 19

5 System Implementation 21

5.1 System Architecture 21
5.2 Environment and Tools Used 22
5.3 Functionality 22
5.3.1 Operate on Opened Web Page 23
5.3.2 Toogle Button Purpose 23
5.3.3 Tagging Web Page Elements 24
5.3.4 Tagging Using Multiple Classes 25
5.3.5 Extract JSON-LD Script 25
5.3.6 Script Validation 25

6 System Testing and Evaluation 28

7 Conclusions 30

References 31

List of Figures

1.1 Problem Description 4

4.1 System Context Diagram 16
4.2 Data Labelling Flow Chart 17
4.3 Flow Chart of Data Annotation System 19
4.4 GUI Design 20

5.1 Toogle view 23
5.2 Click View 24
5.3 Properties Display 24
5.4 Copied Script 25
5.5 Valid Script 26
5.6 Valid Script 26
5.7 Valid Script 27

List of Tables

6.1 Different Test Case Scenarios 29

Acronyms and Abbreviations

SEO	Search Engine Optimization
HTML	Hyper-text Markup Language
JSON-LD	JavaScript Object Notation for Linked Data
CSS	Cascading Style Sheets
PC	Personal Computer
URL	Uniform Resource Locator

Chapter 1

Introduction

1.1 Project Overview

SEO is aimed at increasing organic search traffic referrals to websites. Basically, it is made up of four parts: keyword research, search engine indexing, on-page optimization, and off-page optimization. It is based on the premise that people search with intent rather than allowing probability to lead their interests, and that tracking keyword search referrals can help them find what they're looking for.

To put it another way, SEO works by showing search engines that your content is the top-tier end results for the topic within reach because all search engines have the identical goal: to show your users the best and most relevant results. In order to help search engines to grasp content and represent it in a better way within the search results, a technique is used, known as schema markup. This code can be added to HTML for the websites so that they can provide more detailed and informative results for searchers. Once delivered to the website, schema markup creates a greater specific description in the page. When there is a use of schema markup, Google can simply perceive the web page's content. As a result, the web page gets a rich snippet (Google search result that shows extra information between the URL and the meta description) and may become an entity in Google's data Graph. Basically, it is used to tell Google and other search engines what web page content is about. Thus, Google can give that data to searchers or users and show them the most relevant outcomes.

A complete list of values and items that can be marked up with schema tags is at scheme.org. Schema.org is a way to feature machine-readable facts to web pages that are processed by the main search engines like google to enhance or boost search performance. It is used to create, maintain and promote schemas for structured data on the internet, on web pages, and in email messages. Schema markup is most often used to feature more information about values in mostly these categories that google allows:

- Creative Work
- Event
- Medical Entity
- Organization
- Person
- Place

So, we are going to create a data annotation system by providing the interface/environment as a tool in which annotation of web page elements will be implied using schema.org vocabulary.

1.2 Problem Description

Data annotation is a process in which the data available on site is labeled in multiple formats. To help the machine learning algorithms better identify the data, data annotation is a must. Markup creation for every web page is done manually and similarly if the content on the web page changes, manually updating the markups is required. So doing this for every web page will require a high cost. To save this cost automation is required so machine learning could take place but again, there is a problem that there are no data sets available for it because there is no annotation tool available to create the new data sets. There are certain tools/interfaces available for text annotation, but they do not work with web page-specific content i.e HTML pages, even google structure data markup helper only allows a few classes.

The system we are going to develop will mainly target the two types of audiences that are Researchers and Web Marketers (SEO). This annotation system will help the researchers get the labeled data of any HTML page and so will it help in the process of SEO to bring up the content of the website to the topmost search. Semantic blocks are used for easy selection so they should be kept up to date for better working so it should be kept in mind that the records are up to date. As explained in the below figure there arises a problem where we have to tag the web page elements from a different class.

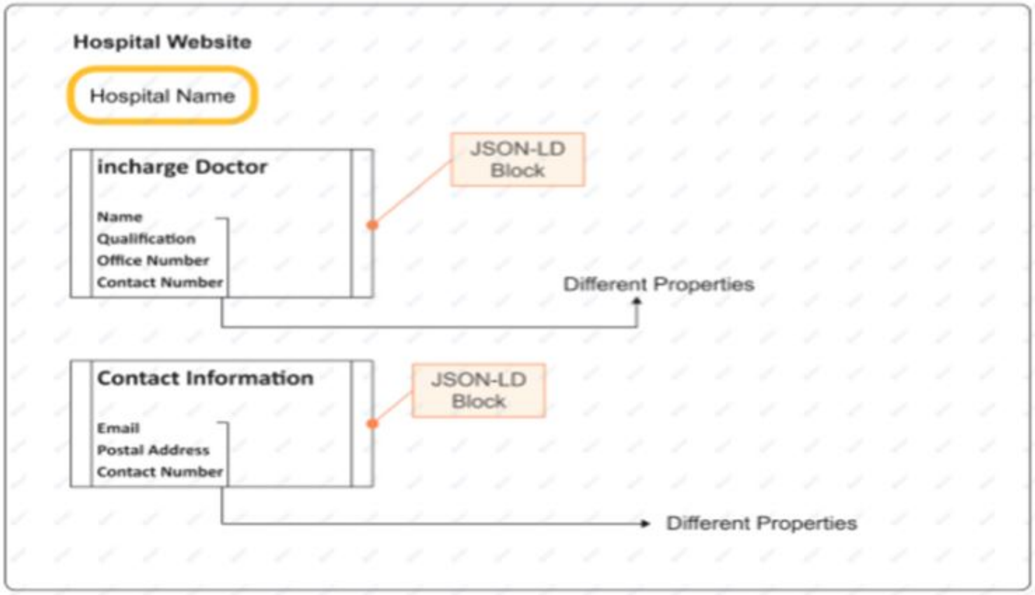


Figure 1.1: Problem Description

Let's suppose there is a hospital website as explained in figure 1.1, which has two different JSON-LD blocks with different properties associated with each block. So here arises the problem of having two different schema.org classes and for this, we are proposing our solution to provide an environment where the user could get the elements tagged having multiple schema.org classes.

1.3 Project Objectives

To design an HTML content annotation system for schema.org based SEO. A tool as a part of an extension would be developed in which the user selects properties for schema.org for its element on the web page. This will help the users to get the script for their web pages which will ultimately help them in process of SEO and data sets formation.

1.4 Project Scope

The scope of this solution is critically important to our main targeted audiences which are researchers who create the data sets for automation purposes and the SEO Web marketers. Different functionalities will be provided to the audience that will use our solution. We will give them an option for choosing the classes of schema.org and then the properties from classes to ease them in their data labeling and also allow them to manually insert values into the tags. Machine learning and data set creation are not included but the environment is provided to help the user easily create that.

1.5 Methodology

A data annotation system for schema.org based SEO is made that is used as a part of the google chrome browser's extension. when we click on the extension button, the extension will pop up and show us the schema classes available. after we click on the desired class, its associated properties will pop up on the screen, and then we could tag keep on tagging the elements and then we click on the copy button, and the created structured data as JSON-LD is copied to our clipboard that can be injected into the head or footer of your site to improve visibility to the search engines. There is going to be a challenge for us to tag the multiple entities at once on a single web page and store them.

Chapter 2

Literature Review

Semantic Web

The Semantic Web, Web 3.0, the web of data or whatever you call it represents the following significant advancement in connecting and representing data. It enables data to be connected from a source to some other source and to be perceived by computers. It is a method for machines to efficiently understand and respond to complex human requests based on their meaning.

Semantic implies understanding, it is concerned with the meaning of the data not the structure of data which is the major difference between the semantic technologies and other technologies related to data. The key objective of the Semantic Web is to set off the advancement of the existing Web to enable users to search, find, share and get data together with less effort. Humans can use the Internet to perform a variety of jobs, including booking tickets in advance, searching for various information, using pronunciation guides, and so on. Nonetheless, machines are unable to perform any of these jobs without the help of individuals, as web pages are designed to be read by humans, not machines. The Semantic Web is a vision for the future in which machines can quickly analyze the data, allowing them to perform a variety of tedious functions such as discovering, integrating, and responding to information available on the Internet.

2.1 Schema.org

Schema.org is one of the primary drivers for a wide range of businesses and individuals to adopt Semantic Web concepts for real-world applications. Schema.org is a structured markup language that lets you specify the sort of content on your website. It helps you to tell search engines that you're talking about a specific person, event, product, new job position, or other categories. It is an approach to enhancing search performance by adding machine-understandable data to web pages that are processed by major search engines. At least for the most common types of Web content, Schema.org has addressed a significant challenge for the Web: making it simple to annotate the data inside web pages. As a result, current applications such as rich snippets can now be used more broadly. Basically, a rich snippet is a service that allows webmasters to add semantics to their pages. Schema.org has become a widely accepted standard for marking up structured data in HTML web pages, Promoted by major search engines. Furthermore, integrating proper Schema into your HTML increases the likelihood of your site being listed as improved snippets in search. Enhanced snippets stand out more than standard snippets and can result in more visits to the website.

2.1.1 How it's Organized:

Bing, Google, Yahoo!, and Yandex formed Schema.org in 2011 to standardize search engine markup. The goal was to create a language that was both short and easy to use, as well as powerful and expressive, for describing "things" on the Web and making them machine-readable and understandable. So far, schema.org/Hotel has vocabulary for expressing the essential data of a hotel, such as its name, address, and description, as well as an email address, a phone number, and special offers. Detailed information, such as the number of beds in a room, the type of bed, whether pets are permitted, and many other detailed data can be considered core data. Schema.org has been extended to allow for greater, more expressive annotations of accommodation data. Its types and its properties can be used to annotate information on various websites in a machine-readable and expressive method.

Schema.org divides data into types and gives a schema for each data type to provide a uniform form of expressing data on websites. Each schema is made up of a set of properties. The schema near the top of the hierarchy is generic, while the schema lower in the hierarchy is more specific. These markups allow search engines to see the meaning and relationships behind entities mentioned on the website. Once added to a web page, schema markup creates an enhanced description that appears in search results.

2.1.1.1 Types and Properties of schema.org:

The schema near the top of the hierarchy is generic, while the schema lower in the hierarchy is more specific making a hierarchical flow of types. The vocabulary currently consists of 797 Types, 1453 Properties 14 Data types, 86 Enumerations, and 462 Enumeration members.

Followings are some commonly used classes and their certain proprieties some of which are also used in our plugin:

2.1.1.2 Organization:

It has various classes some of which are:

- address
- email
- faxNumber
- hasCredential
- location
- telephone

2.1.1.3 Person:

It has various classes some of which are:

- additionalName
- address
- affiliation
- email
- gender
- jobTitle

2.1.1.4 Book:

It has various classes some of which are:

- abridged

- bookEdition
- bookFormat
- illustrator
- isbn
- numberOfPages

2.2 Schema Validator

Schema has a validator that serves the purpose of script validation. It is one of the most authentic validators and it can either fetch URLs or use code snippets to validate the script. The script is pasted in the code snippet and then validated. If the results show no error or warning it means the script is validated and it can be used in your custom website to improve your search rankings.

2.3 Data Annotation System

Data annotation is the task of labeling or identifying data in various formats like text, images, and videos. Basically, it comes down to labeling the area or region of interest, this type of annotation is found specifically in images and videos. On the other hand, annotating text data largely encompasses adding relevant information, such as metadata, and assigning them to a certain class. In machine learning, the task of data annotation usually falls into the category of supervised learning where the learning algorithm associates input with the corresponding output and optimizes itself to reduce errors.

There are several tools available for annotation of text, images, and videos which include:

- Label studio [1] for including image classification, object detection, semantic segmentation.
- Diffgram [2] for spatial annotation in both image and video.
- ImageTagger [3] for image sets to be labeled with bounding boxes, polygons, lines and key points.

For Web page annotation there are certain tools available that have their limitations and uses that includes some of the tools are:

- Google Structure markup Helper [4] which allows users to deal with twelve classes of schema.
- Merkle [5] which allows users to deal with four classes and generate JSON-LD script for them.
- RankRanger [6] which allows users to deal with three classes and generate scripts against them.

These data annotation techniques help the researchers and Web Marketers, Benefit them in acquiring the data sets they want in less time with a user-friendly interface to automate.

[1] https://labelstud.io/

[2] https://diffgram.com/main/

[3] https://imagetagger.bit-bots.de/

[4] https://www.google.com/webmasters/markup-helper/

[5] https://technicalseo.com/tools/schema-markup-generator/

[6] https://www.rankranger.com/schema-markup-generator

Chapter 3

Requirement Specifications

Proposed System

We are going to create a data annotation system by providing the interface/environment as a tool in which annotation of web page elements will be implied using schema.org vocabulary. It will benefit researchers in acquiring the data sets they want in less time with user friendly interface to automate whereas SEO is the main functionality in web marketing that would be getting benefit from our solution to promote their web content and make the results prior in search engines with the help of labeled tags.

User will be able to tag multiple entities on a single web page at the same time hence saving cost and time. We will give them an option for choosing the classes of schema.org and then the properties from classes to ease them in their data labeling and use the multiple classes and their properties which are given or specified.

3.1 Requirement Specifications

Machine learning and data set creation are not included but the environment is provided to help the user easily create that. For this, we have made an extension in chrome browser and for it to work efficiently we need google chrome, an active internet connection and proper use of file handling.

3.1.1 Functional Requirements:

- The system should be able to tag the web page elements accurately.
- The system should be able to generate valid JSON-LD for the tagged information.
- The system should provide the user to be able to tag multiple entities at once.
- The user could add whatever class he desires from schema.org.
- The user could manually input the value for tags as well.
- The user will be able to download and copy the JSON-LD script.
- The user could export annotated file for use as a data-set.

3.1.2 Non-Functional Requirements:

- The system should save time of the user by allowing him to use multiple classes at a same time.
- The system should be easy to use making it easier for users to use the plugin.

- The authenticity of the system should be better that is the script generated would be a valid script.

Chapter 4

Design

4.1 System Architecture

The system operates on google chrome as a part of extension which will open its interface on click and then it will allow the user to choose between the classes he wants to use at first and then after choosing a class user can tag the properties associated with that class.If the user wants to use more than one class then he's allowed to do so by clicking right click on the screen and there he can click on the name of class he wants and it will be added. The system will at the end generate a JSON-LD script to the user once he clicks on the copy button and the generated JSON-LD script is copied into the clipboard that can be injected into the head or footer of your site to improve visibility to the search engines. As explained in above Figure 4.1 ,When the plugin is clicked, It allows user to choose the class to start tagging web page elements according to the properties defined in the class and then generate the extracted JSON-LD script.

4.1.1 System Context Diagram

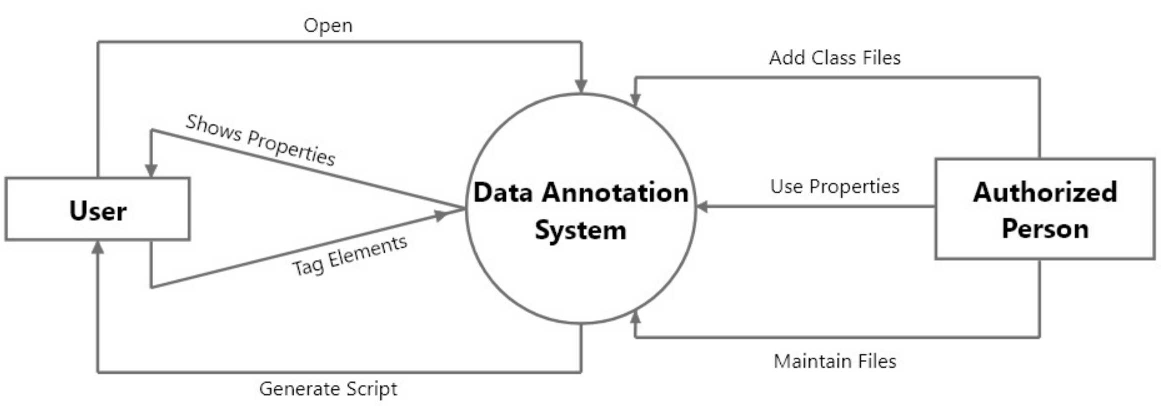


Figure 4.1: System Context Diagram

As shown in the above figure 4.2, A user is allowed to open the plugin and see the properties of classes as well as the classes given and then tag the elements of web page to generate script at the end Where as the owner or authorized person of the plugin is able to add new class files along with their properties in the Plugin and is responsible for main ting those files too.

4.1.2 Labeling Procedure

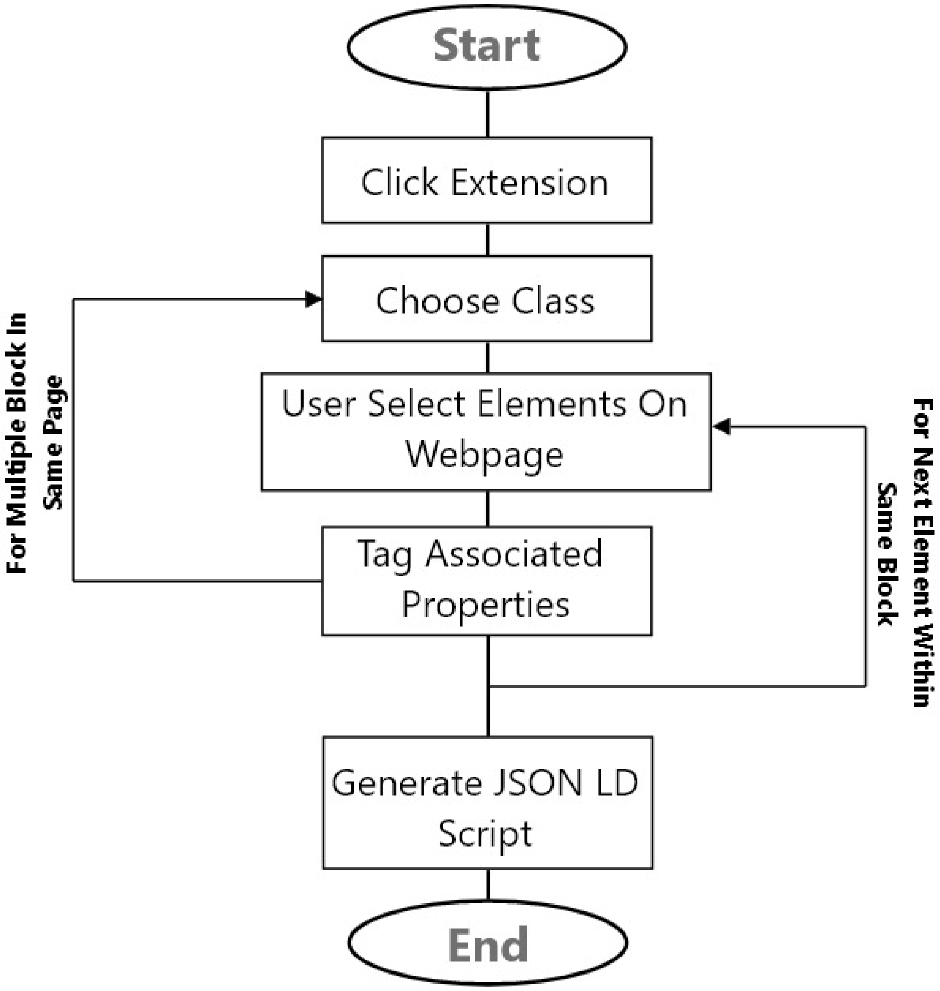

Start
Click Extension
Choose Class
User Select Elements On Webpage
Tag Associated Properties
Generate JSON LD Script
End
For Multiple Block In Same Page
For Next Element Within Same Block

Figure 4.2: Data Labelling Flow Chart

As explained in the above figure 4.3, when a user starts tagging the web page elements, he can choose the class and then start tagging the data with respect to the properties of the class and he keeps on tagging the elements but if he wants to use another class at the same time , he will choose another class and it’s properties will be shown to the user and then he can tag those elements too being on the same page without going back to start and when the user finishes tagging the elements he could simply click to generate the JSON-LD script he created.

4.2 Design Constraints

There are certain constraints in the system that is being developed that, it will only target the static pages and not the dynamic pages such as social media websites because there is a continuous change in content on such websites and it also a security issue.It will only be able to tag the web pages that can use the class we have defined for example we open YouTube and then want to tag elements, we don't have a class and its property for it to do so. The interface of the plugin changes its font size with respect to the web page that is opened.

4.3 Design Methodology

A data annotation system for schema.org based SEO is made that is used as a part of google chrome browser's extension. when we click on the extension button, the extension will pop up and show us the schema classes available, after we click on the desired class, it's associated properties will pop up on the screen and then we could tag keep on tagging the elements and then we click on copy button and the created structured data as JSON-LD is copied to our clipboard that can be injected into the head or footer of your site to improve visibility to the search engines or even it could be downloaded in a file if user wants to export it into the system he is using. There is going to be a challenge for us to tag the multiple entities at once on a single web page and store them making it easier for users to not to generate scripts separately for each class.

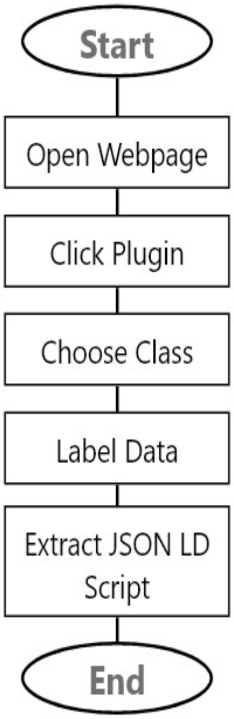


Figure 4.3: Flow Chart of Data Annotation System

4.4 GUI Design

The graphical user interface of our extension is going to be user friendly and interactive.The user interface of this extension looks like what any user would expect a modern, trusted extension to look like; clean, simple, elegant and efficient.

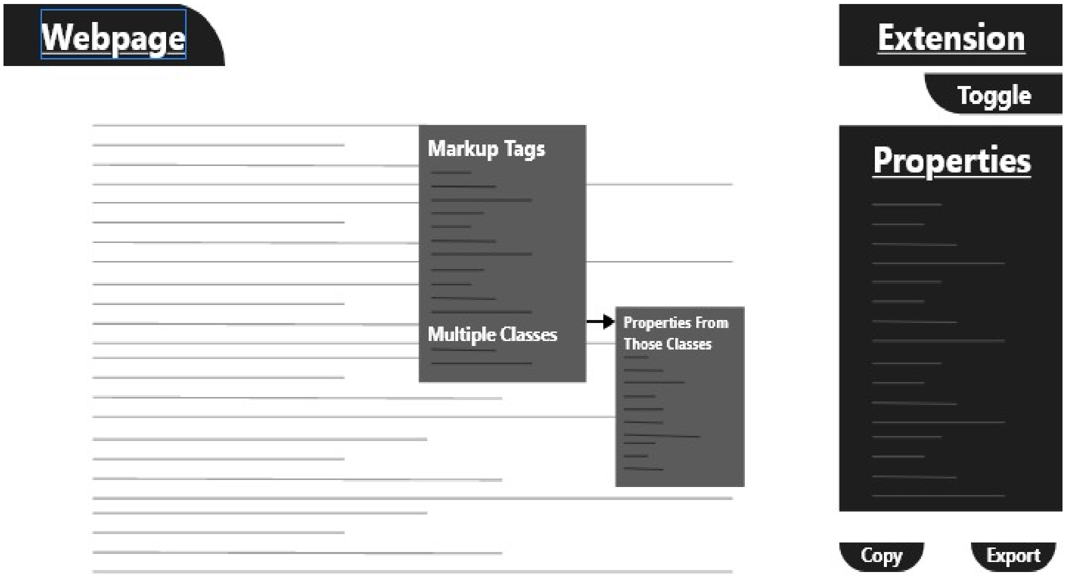


Figure 4.4: GUI Design

As shown in figure 4.5 design, When a user opens the web page and then clicks on the extension button it will a menu of list will appear where a user will be able to see the classes he can use and then we he will choose the class a toogle button will appear on the top right side of the web page and when a user will click on that button, a list of properties of the class he choose will appear on the right side of the page below toogle button. when User starts to tag the web Page elements he will be able to see those properties in the click when he click on the web page to tag elements and in that view user will be able to see other classes as well so if he wants to use them too, he click on the class name and properties of that class will also appear on right side of the page below the other class properties and then he could tag the web page elements against that class as well and then he can finally click on copy button to copy the extracted JSON-LD scripts in his keyboard or clicks on export button to download it in a file.

Chapter 5

System Implementation

Implementation is the process of moving an idea from concept to reality. In this section we have explained the detailed implementation and functionality of our extension.

5.1 System Architecture

Our system that we have created is an extension which is used as a part of google extension to make it use simply and having a user interactive interface. When clicked on the extension it will show a list of classes that we have used in our program and then amongst those classes we could choose the class we want to use. Once a class is used, a user could use more classes being on the same page without going back, he clicks on the screen and choose the class name he wants to use.

In this way user is able to use more than just one class for schema markups and at the end when he is done, he could simply click on the copy button to save the generated JSON-LD script in to the clipboard for further use.

5.2 Environment and Tools Used

The first step was to create a manifest file named manifest.json which is a metadata file in JSON format that contains properties like extension's name, description, version number and so on. In this file we tell Chrome what the extension is going to do, and what permissions it requires. After making this file we created further CSS and HTML files and loaded our extension in chrome by turning on the developer mode.

We compiled the program in Microsoft Visual Studio which is an effective and fast platform for programming. We used HTML/CSS, JavaScript, Jquery along with file handling to make our plugin task oriented and our functions to perform effectively.

5.3 Functionality

The functionality of our plugin works in a way that it provide the environment where user is allowed to use the schema markups and tag the web page elements they desire and then create a JSON-LD for it at the end. Once the User clicks on the plugin it will show the classes available for which user could use schema markup tags and then when he choose the class, he will get the properties related to that class shown on the right side of the page which could be toggled, he can use those properties and tag the web page elements.

Provided user could even use multiple classes at a single time right when he clicks on the screen to tag web page elements, he will see the class name there which is available and once he selects the class, all of its properties are shown in the plugin and then he could use their properties as well and tag the elements on opened web page.

5.3.1 Operate on Opened Web Page

The plugin that we have been able to develop works on the web page that is currently opened in background. When we click on the extension we are able to tag the choose the class and further tag the web page element on the same page that was opened.

5.3.2 Toogle Button Purpose

when the user click on the extension it will show the user a list of classes in a menu out of which a user can select a class and then a button on top right saying toogle will appear.It is used to show and hide the list of properties for that class for a better view of the web page as show in figure 5.1.

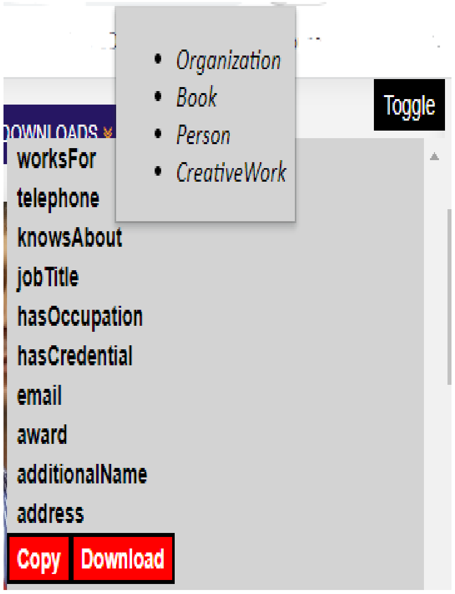


Figure 5.1: Toogle view

5.3.3 Tagging Web Page Elements

The user can start tagging the web page elements once he selects the class to start with. When a user selects the element on web page a view is displayed as shown in figure 5.2.

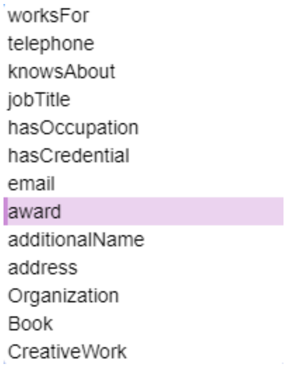
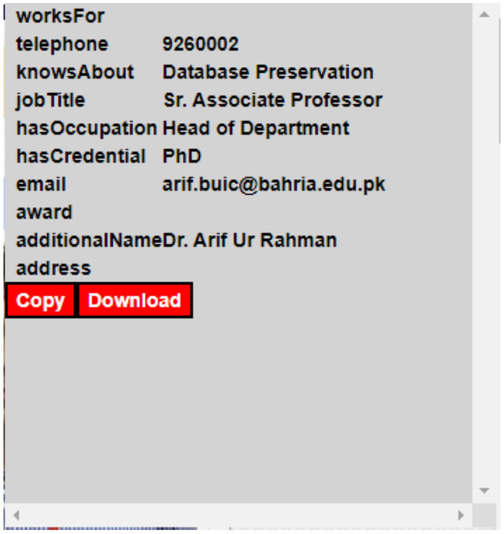


Figure 5.2: Click View

When a User tag all the elements all the tagged properties are showed on the right side of the web page as shown below in figure 5.3.



worksFor	
telephone	9260002
knowsAbout	Database Preservation
jobTitle	Sr. Associate Professor
hasOccupation	Head of Department
hasCredential	PhD
email	arif.buic@bahria.edu.pk
award	
additionalName	Dr. Arif Ur Rahman
address	

Figure 5.3: Properties Display

5.3.4 Tagging Using Multiple Classes

when a user wants to use multiple classes then when he selects the first class in plugin and toogle button appears. When user click on the screen then there he could see the other class names he want to use in a click view as shown in figure 5.2.
When the user click on the other class name in click view then user is able to see the properties of this new added class on right side in plugin view beneath toogle button.

5.3.5 Extract JSON-LD Script

When the user is done tagging the elements on the web page then he generate the JSON-LD script which could either be downloaded in a file or directly be copied to the clipboard for utilizing the script he generated for rich results.

bahria.edu.pk says

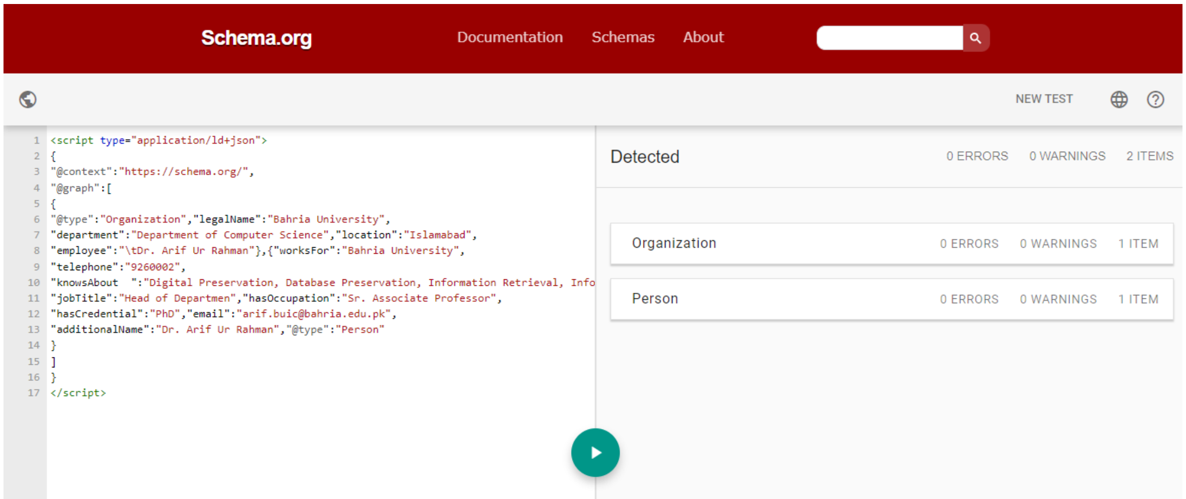
Copied to Clipboard

Figure 5.4: Copied Script

When the user clicks in the copy button on the bottom of the plugin view as you can see in figure 5.3 then a pop up of script copied to clip board is displayed as shown in above figure 5.5. User may also be able to download the script by clicking on the download button.

5.3.6 Script Validation

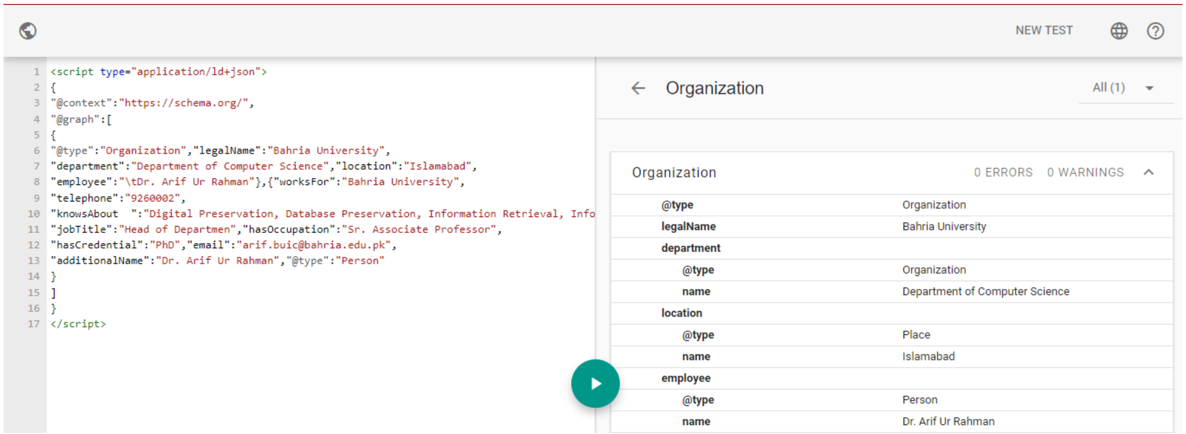
A user can even go to schema validator to check if the script he generated is valid or not and if the script is not valid or have any errors. It will tell the user that the script is having warning or errors in script but if valid , it shows the script is valid by giving no errors and warnings as shown in figure 5.6 below.



```
<script type="application/ld+json">
{
"@context":"https://schema.org/",
"@graph":[
{
"@type":"Organization","legalName":"Bahria University",
"department":"Department of Computer Science","location":"Islamabad",
"employee":"\tDr. Arif Ur Rahman"},{"worksFor":"Bahria University",
"telephone":"9260002",
"knowsAbout  ":"Digital Preservation, Database Preservation, Information Retrieval, Info
"jobTitle":"Head of Departmen","hasOccupation":"Sr. Associate Professor",
"hasCredential":"PhD","email":"arif.buic@bahria.edu.pk",
"additionalName":"Dr. Arif Ur Rahman","@type":"Person"
}
]
}
</script>
```

Detected	0 ERRORS	0 WARNINGS	2 ITEMS
Organization	0 ERRORS	0 WARNINGS	1 ITEM
Person	0 ERRORS	0 WARNINGS	1 ITEM

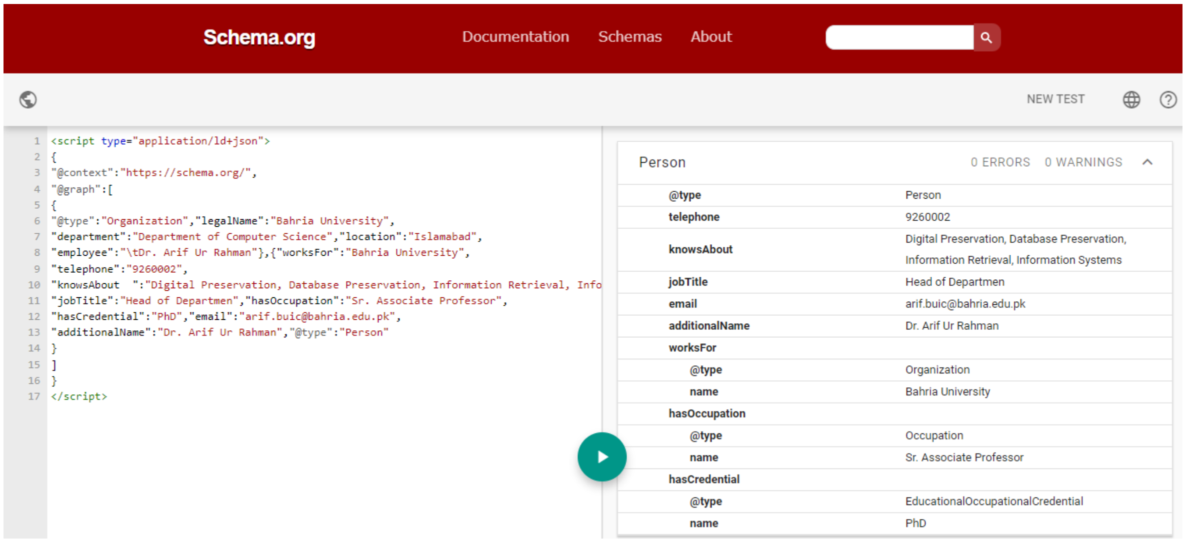
Figure 5.5: Valid Script



```
<script type="application/ld+json">
{
"@context":"https://schema.org/",
"@graph":[
{
"@type":"Organization","legalName":"Bahria University",
"department":"Department of Computer Science","location":"Islamabad",
"employee":"\tDr. Arif Ur Rahman"},{"worksFor":"Bahria University",
"telephone":"9260002",
"knowsAbout  ":"Digital Preservation, Database Preservation, Information Retrieval, Info
"jobTitle":"Head of Departmen","hasOccupation":"Sr. Associate Professor",
"hasCredential":"PhD","email":"arif.buic@bahria.edu.pk",
"additionalName":"Dr. Arif Ur Rahman","@type":"Person"
}
]
}
</script>
```

Organization	0 ERRORS 0 WARNINGS
@type	Organization
legalName	Bahria University
department	
@type	Organization
name	Department of Computer Science
location	
@type	Place
name	Islamabad
employee	
@type	Person
name	Dr. Arif Ur Rahman

Figure 5.6: Valid Script



```
<script type="application/ld+json">
{
"@context":"https://schema.org/",
"@graph":[
{
"@type":"Organization","legalName":"Bahria University",
"department":"Department of Computer Science","location":"Islamabad",
"employee":"\tDr. Arif Ur Rahman"},{"worksFor":"Bahria University",
"telephone":"9260002",
"knowsAbout  ":"Digital Preservation, Database Preservation, Information Retrieval, Info
"jobTitle":"Head of Departmen","hasOccupation":"Sr. Associate Professor",
"hasCredential":"PhD","email":"arif.buic@bahria.edu.pk",
"additionalName":"Dr. Arif Ur Rahman","@type":"Person"
}
]
}
</script>
```

Person 0 ERRORS 0 WARNINGS

@type	Person
telephone	9260002
knowsAbout	Digital Preservation, Database Preservation, Information Retrieval, Information Systems
jobTitle	Head of Departmen
email	arif.buic@bahria.edu.pk
additionalName	Dr. Arif Ur Rahman
worksFor	
@type	Organization
name	Bahria University
hasOccupation	
@type	Occupation
name	Sr. Associate Professor
hasCredential	
@type	EducationalOccupationalCredential
name	PhD

Figure 5.7: Valid Script

Chapter 6

System Testing and Evaluation

The proper testing of our plugin is done in this section which has its own certain limitations and as well as successes factors. Testing is a crucial part of a project which shows how effective a project is and what are its limitation. Testing of our plugin is done with respect to certain cases like test case type, description, test step, expected result and Status.Below is a table given explaining different test scenarios and their results accordingly.

Test ID	Test Case Type	Description	Test Step	Expected Result	Status
01	Graphical User Interface	The user Interface of the plugin should remain same on all websites	Open plugin on different websites	The layout of plugin would be same on all of website	FAIL
02	Usability Testing	User will successfully tag the web page elements	open a website and click on the plugin to choose a class to start tagging web page elements	User should be able to tag web page elements against the defined properties of class he choose	PASS
03	Functionality Testing	User will copy the generated JSON-LD script for tagged web page elements	Tag the elements on web page according to the properties defined	User will be able to copy the generated script when he clicks on the copy button after tagging the web page elements	PASS
04	Performance Testing	The user will check the performance/accuracy of the plugin to see if the script he generated is even valid or no	The generated JSON-LD script is pasted in a schema validator	The validator should validate the pasted script without any error or warning	PASS
05	Compatibility Testing	The plugin is compatible with browser other than google chrome installed on a PC	Open another browser to see if the plugin works	The plugin should work on every browser	FAIL

Table 6.1: Different Test Case Scenarios

Chapter 7

Conclusions

The use of use of JavaScript and Jquery in this project helped us to achieve what we wanted to develop. We aimed to build this project because , In order to automate this manual markup creation for every web page the automation requires machine learning which could not be possible without having data sets. So In order to create new data sets we needed an annotation tool to provide the environment to such users which are willing to automate this in future and perform machine learning operations using our tool by creating the data sets. So this project will help user in process of creating data sets for their automation procedure. It was tricky to use the multiple classes at all once because no such tool exists in the market today that even google's owned schema markup helper only deals with one class at a time.

References

- https://schema.org/docs/schemas.html

- https://www.impactplus.com/blog/what-is-schema-markup-and-how-to-implement-it

- https://moz.com/learn/seo/schema-structured-data

- https://developers.google.com/search/docs/advanced/structured-data/intro-structured-data

- https://www.v7labs.com/blog/data-annotation-guide

- https://technicalseo.com/tools/schema-markup-generator/

- https://www.rankranger.com/schema-markup-generator

- https://www.contentkingapp.com/academy/schema/

- https://www.techopedia.com/definition/27961/semantic-web

- https://cambridgesemantics.com/blog/semantic-university/intro-semantic-web/

- Hepp, Martin. "The web of data for e-commerce: Schema. org and GoodRelations for researchers and practitioners." In International Conference on Web Engineering, pp. 723-727. Springer, Cham, 2015.

- Patel-Schneider, Peter F. "Analyzing schema. org." In International Semantic Web Conference, pp. 261-276. Springer, Cham, 2014.

- Mika, Peter. "On schema. org and why it matters for the web." IEEE Internet Computing 19, no. 4 (2015): 52-55.

- Hop, Walter, Stephan Lachner, Flavius Frasincar, and Roberto De Virgilio. "Automatic web page annotation with Google rich snippets." In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", pp. 957-974. Springer, Berlin, Heidelberg, 2010.

- Meusel, Robert, Christian Bizer, and Heiko Paulheim. "A web-scale study of the adoption and evolution of the schema. org vocabulary over time." In Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics, pp. 1-11. 2015.

- Kärle, Elias, Umutcan Simsek, Zaenal Akbar, Martin Hepp, and Dieter Fensel. "Extending the schema. org vocabulary for more expressive accommodation annotations." In Information and Communication Technologies in Tourism 2017, pp. 31-41. Springer, Cham, 2017.