

Web Page Generator: an Online Tool for Supporting Web Programming Courses

Carlos R. Jaimez-González, Alfredo R. Vargas-Rodríguez

Departamento de Tecnologías de la Información, Universidad Autónoma Metropolitana - Cuajimalpa, Av. Constituyentes No. 1054, Col. Lomas Altas, C.P. 11950, México D.F.
{cjaimez,207363956}@correo.cua.uam.mx

Abstract. This paper presents Web Page Generator, which is an online tool for supporting Web Programming courses. Users of Web Page Generator can create a web page graphically, by placing web elements on a work area; they can also modify the style of the web page through an edition bar, and add validations to input fields. Additionally, users can navigate the web page structure using a tree bar, and visualizing it in a preliminary view. The resulting web page is generated in three separated files: an HTML file for its structure, a CSS file with styles for its presentation, and a JavaScript file for its validations. Web Page Generator allows web developers producing web pages, but also supports teaching activities in courses related to web programming.

Keywords. Web page generator, web programming, automatic code generation, html, javascript, CSS, web page, WYSIWYG.

1 Introduction

The creation of web sites in recent years has been increasing, given the popularity of the Internet to allow advertising products and services with no restrictions of time, 24 hours a day, 365 days a year. This situation opens the need for tools that allow the development and automatic generation of HTML code, so web sites can be implemented rapidly and efficiently. Web developers have had to adapt to the existing development environments, which are sometimes complicated and difficult to understand; some of these need some training previous to their use [1], [2]; and some others are very expensive [1]. A similar problem can be found in some undergraduate degrees, in which there are courses where students are introduced to web site development and the use of graphical tools. This is the case of two undergraduate degrees at our faculty: Design, and Information Technologies and Systems, in which students take courses where they are required to create web pages, such as Web Design, Web Pages Development, Web Programming, etc.

This paper presents Web Page Generator, an online tool for creating web pages and generating HTML, CSS and JavaScript code automatically. This tool has been created not only for web developers, but also to support teaching activities in courses related to web programming. One of the key advantages of this online tool is the clear dis-

tion of HTML, CSS, and JavaScript code, which is generated in separated files. The work presented in this paper is based on the initial architecture described in [3].

The rest of the paper is organized as follows. Section 2 presents some existing tools that are relevant to this development; mainly related to code generators by conceptual models, and generators by edition. Section 3 describes the architecture of the online tool, its modules and their functionality. The web user interface of the tool and its components are presented in section 4. Some examples of web pages generated with Web Page Generator are presented in section 5. Finally, section 6 provides conclusions and future work.

2 Existing generation tools

There are some online tools that generate HTML, but their functionality is limited [4], [5]. Some others also generate HTML code, but also allow restructuring the data that is the source of the web page, the generation by conceptual models or customization, and modification from web pages already generated. Given the different objectives of these code generators, they are classified in two types: 1) generators by conceptual models or customization; and 2) generators by edition. The following sections present these types of generators, as well as the problems in these areas.

2.1 Tools based on conceptual models

The tools that work based on conceptual models generate a kind of platform that can be used by developers to produce web content through an easy-to-use and customizable interface. This is the case of the Web Interface Development Environment (WIDE) [6], which provides automatic generation of web content, starting from a conceptual model. To facilitate the construction of web content, this tool incorporates a layers system, which allows the visualization of control and its functionality; and also has a zoom system to increase the level of detail in the production of content.

Another tool based on the development by conceptual models is the Web Application Rapid Prototyping (WARP) [7]. This application offers a set of online software tools that help the designer in the navigation of a web application, using the UWA methodology [8]. GIWA is another tool based on conceptual models, which provides a generator of adaptive web applications [9]. Its main objective is to facilitate the design and automatic generation of web interfaces through several levels, which go from functionality to presentation.

2.2 Tools based on edition

The edition tools work on existing content and aim to have an easy web site administration and upgrade. The administration goes from the web site presentation to the modification of its source data. Nakano [10] developed a tool that edits a web site based on a bidirectional transformation, with an upgrade system called Vu-X. This way, users modify directly the web page, without the need of accessing the database;

and the result is shown automatically in the database using the bidirectional transformation language Bi-X [11]. Vu-X also implements an editing system similar to those called What You See Is What You Get (WYSIWYG). There is also the tool called Dido [12], which allows editing web pages; it contains an interactive AJAX interface, a data editor, and a MetaEditor that allows WYSIWYG editing. AJAX is the acronym of Asynchronous JavaScript and XML, which is a web development technique used to create interactive applications or Rich Internet Applications (RIA). Dido does not need installation because it works on browsers; this way author documents can share information and at the same time edit it according to their needs. It is limited to certain functionality, such as create, read, update, and delete. WebSheets [13] is another tool that allows the creation and modification of HTML pages with dynamic content; it modifies the database and the structure of the web pages with a WYSIWYG editor.

The tools described in this section are dedicated to the modification of HTML and XML code; but there are also other tools that are aimed to the creation of styles, such as the XSLbyDemo [14] tool, which generates XSL stylesheets from an HTML page, using a WYSIWYG editor. XML is a markup language created by the World Wide Web Consortium (W3C); it derives from SGML, and allows defining the grammar of specific languages (it should be noticed that HTML is also derived from SGML). The eXtensible Stylesheet Language (XSL) is an extensible language based on stylesheets. These languages are part of a family of languages based on XML, which allow describing the information contained in an XML document, and transform it or format it, in order to be presented in a digital media. One of the main features of XSLbyDemo is the generation of a stylesheet from a well-formed XHTML document. The produced XSL file can be associated to several documents derived from the same Document Type Definition (DTD), which describes the rules for the structure and syntax of XML or SGML documents. XSLbyDemo is based mainly in insertions, modifications, copies and elimination of rules that are defined in the file. The generation of that file is carried out through the Document Object Model (DOM) for accessing and manipulating the objects contained in it. DOM is an application programming interface that provides a standard set of objects to represent XML and HTML documents; it is a standard model to access and manipulate these objects.

3 Architecture

Web Page Generator allows the graphical edition of web pages, and generates automatically HTML, CSS and JavaScript code. It uses a WYSIWYG editor, and it is accessible through the web. When the user finishes editing the web page, the result is a set of files with the programming code of the technologies mentioned. In the following sections it will be described the modules that are part of this tool.

The online tool uses a client-server architecture, which is shown in Figure 1. The client side is composed of two blocks: the user interface, and the modules that support it; while the server side is composed of the code generators for HTML, CSS and JavaScript, and the module for generating files. The rest of this section provides a detailed explanation of the components of this architecture.

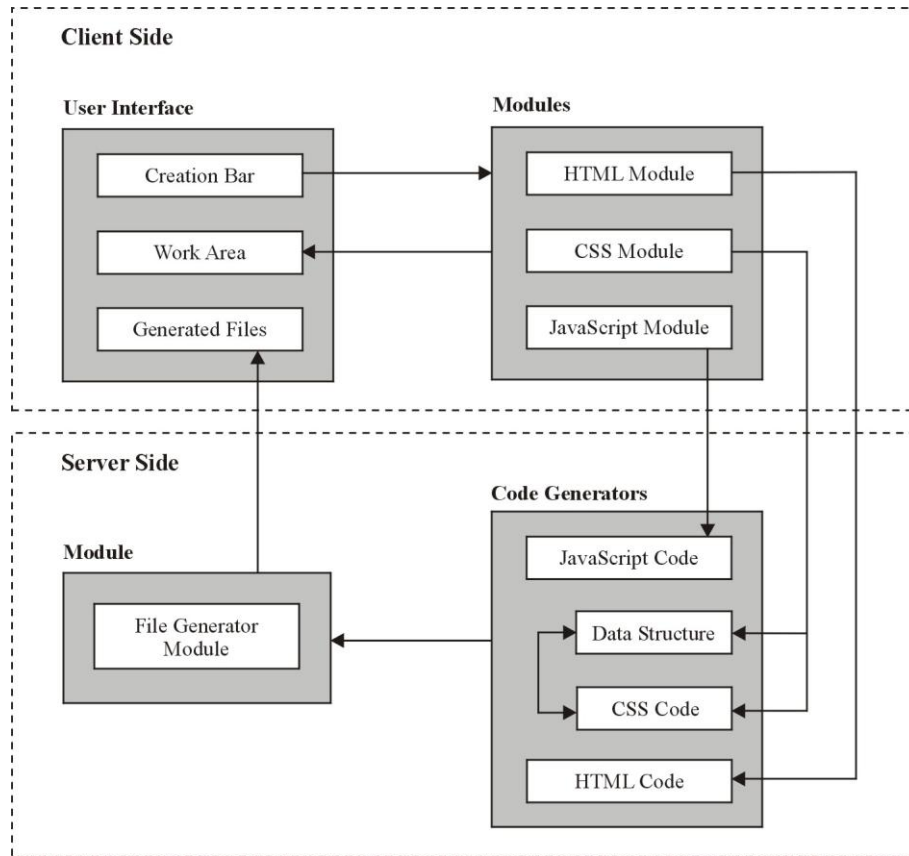


Fig. 1. Web Page Generator client-server architecture.

3.1 Client side

The client side of the tool has two main blocks, which are shown in Figure 1. The first block involves a set of modules for the generation of HTML, CSS and JavaScript code. The second block involves a set of components for the web interface, such as the creation bar, the work area, and the component for the generated documents. The first block is described in the following paragraphs.

HTML Module. This module works on the client side. The user of the tool can place HTML elements on the work area, through the creation bar. This module is also in charge of administering the attributes of the elements used.

CSS Module. This module works mainly on the client side, but it also has part of its functionality on the server side. It allows the creation of CSS styles, which are applied to the HTML elements placed on the work area. It also stores a data structure for the styles created, which are then used for the generation of the code. The tool is able to apply CSS styles at three different levels: 1) *Class level* allows creating a CSS class, defining attributes and values, and applying that class to any element; 2) *Ele-*

ment level allows applying a style to all the elements of the same type on the work area (for example, all div elements on the work area); 3) *Element id level* allows defining a style for a unique element, which is the element selected on the work area.

JavaScript Module. The functionality of this module is on the client side. It allows providing the validation for HTML input elements, where the validation of such elements can be seen instantly on the work area.

3.2 Server side

There are two blocks that are part of the server side, which are illustrated in Figure 1: the code generators and the file generator module. These blocks are described below.

Code Generators. There are three code generators: one for HTML, one for CSS, and one for JavaScript; which generate the necessary code to pass it to the file generator module. The CSS code generator uses a data structure to store in memory all the CSS styles that were created in the CSS module, described previously. The aim of the data structure is to contain, modify or delete all the CSS style information.

File Generator Module. The functionality of this module is divided in two steps. In the first step, it collects the information from the HTML, CSS and JavaScript code generators. In the second step, once the tool has loaded in memory all the code from the three code generators, it applies the format and constructs all the necessary files. It then takes the created files and zips them; finally, a link is returned to the web user interface, where the files generated can be downloaded.

4 Web user interface

The web user interface for the online tool is shown in Figure 2, where its components are identified by numbers: 1) creation bar, 2) work area, 3) edition bar, 4) tree bar, and 5) visualization bar. These components are described in this section.

Creation Bar (1). This bar shows all the HTML elements that can be placed on the work area. The user clicks on the icon of the HTML element desired in order to place it on the work area where the cursor is located. Table 1 illustrates some of the icons used on the creation bar, and the actions associated with them.

Work Area (2). The work area shows the construction of the web page, the HTML elements placed, the CSS styles associated with the elements, the validations for the text fields, and in general any element that is created. The work area has some editor features, so that the cursor can be placed anywhere on it, and write directly with no need of introducing any HTML element explicitly through the bar. Figure 3 shows the work area of the Web Page Generator, with some HTML content placed graphically on it by the user, such as tables with several rows and cells, images, formatted text, hyperlinks, div containers, paragraphs, headers, etc. Additionally, in section 5 there are some examples of completed web pages generated with this tool.

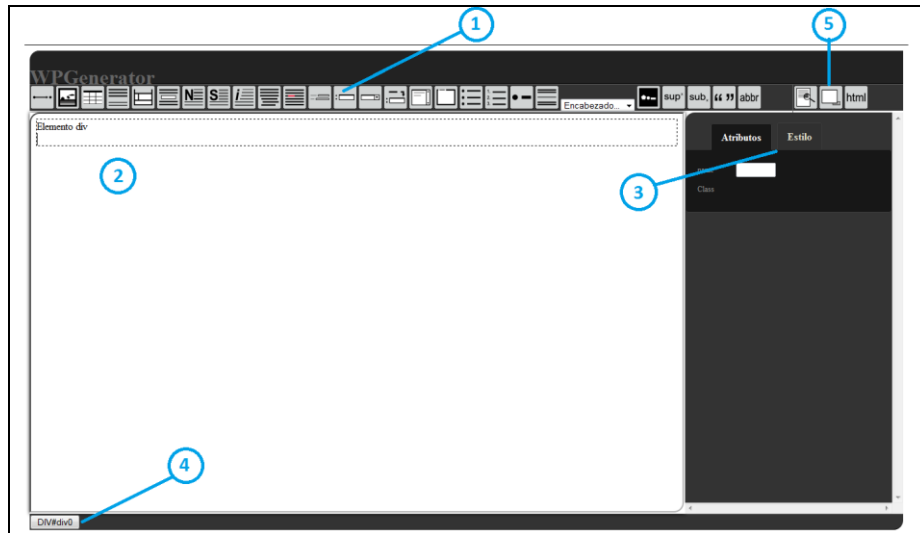


Fig. 2. Web Page Generator user interface.

Edition Bar (3). There are two important tasks that can be accomplished with the edition bar, through the two tabs that appear on it. The *Attributes* tab shows the attributes used on the selected HTML element; and the *Styles* tab shows all the CSS styles that can be applied to the HTML element selected, in different levels: *class level*, *element level*, or *element id level*.

Table 1. Some icons of the creation bar and their associated actions.

Icon	Action	Icon	Action
	It places an element <i>a</i> on the work area.		It shows options for placing an element <i>img</i> on the work area.
	It shows options to place an element <i>table</i> on the work area.		It places an element <i>p</i> on the work area.
	It places an element <i>div</i> on the work area.		It places an element <i>iframe</i> on the work area.
	It places an element <i>form</i> on the work area.		It places an element <i>select</i> on the work area.
	It places an element <i>textarea</i> on the work area.		It places an element <i>ol</i> on the work area.

Tree Bar (4). This bar shows information about the elements tree in the web page being generated. It shows how nested an HTML element is with respect to other HTML elements. It is also used as a selector of HTML elements. It should be noticed that the structure of HTML is based on nesting elements.

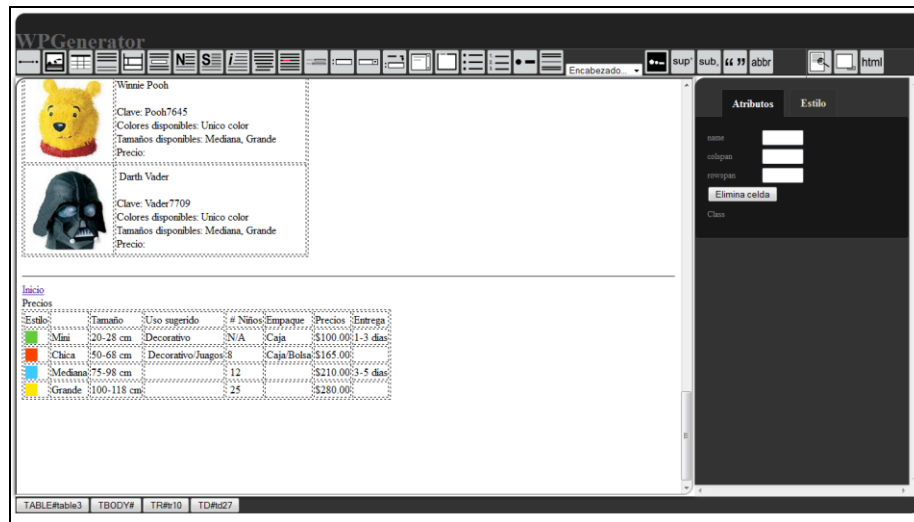


Fig. 3. Work area with some HTML content placed graphically.

Visualization Bar (5). This bar contains three buttons, which are the following (from left to right): *Preview*, *Hide Panel*, and *Generate Code*. The *Preview* button allows visualizing a preliminary view of the web page in full screen; the *Hide Panel* button hides or visualizes the *Edition Bar*; the *Generate Code* button allows generating the three files (with the HTML, CSS and JavaScript code), which are the result of the web page created on the work area. The *Generate Code* button also generates a zip file with the files mentioned, which can be downloaded from the web interface. Figure 4 shows a very small fragment of the HTML code generated automatically (Darth Vader image and text) for the web page illustrated in Figure 3. For display purposes and due to the lack of space, the code is shown without break lines.

```
<tr id="tr8">
  <td id="td16">  </td>
  <td id="td17"> <span id="span40"
    style="font-weight: bold;">Darth Vader</span><br/><br/>
  <!-- rest of the code omitted -->
```

Fig. 4. Fragment of the HTML code generated automatically.

5 Examples of web pages generated

This section shows two examples of web pages that have been generated with Web Page Generator, which has been in constant use. Figure 5 shows a web page with nested HTML elements, which have CSS styles. The web page was generated using div elements, tables, rows and cells, text formats, headers, images and a form with JavaScript validation. It can be observed the use of CSS styles to modify the presentation of the web page, such as the background color at different levels, text color for different headers and text on the page, some other text attributes, table borders, cell borders, text alignment, etc. JavaScript code is only used for validation of the input HTML elements of the form, at the end of the web page.



Fig. 5. Web page generated with Web Page Generator.

Figure 6 illustrates a web page with a more complex distribution of elements, background images, a menu, and text formats. It also uses div elements to separate the content of the web page; it has a banner on top, a top menu, a right menu, and the content aligned to the left. Some of the HTML elements used are the following: lists, tables, images, forms, inputs, paragraphs, headers, hyperlinks, divs, etc. The use of CSS styles to modify the presentation of the web page is also illustrated with color attributes, text attributes, table attributes, div attributes, paragraph attributes, and some other CSS attributes. Concerning the JavaScript programming language, the web page generated in this example, also includes a form with input elements and validations on them, such as the email format.

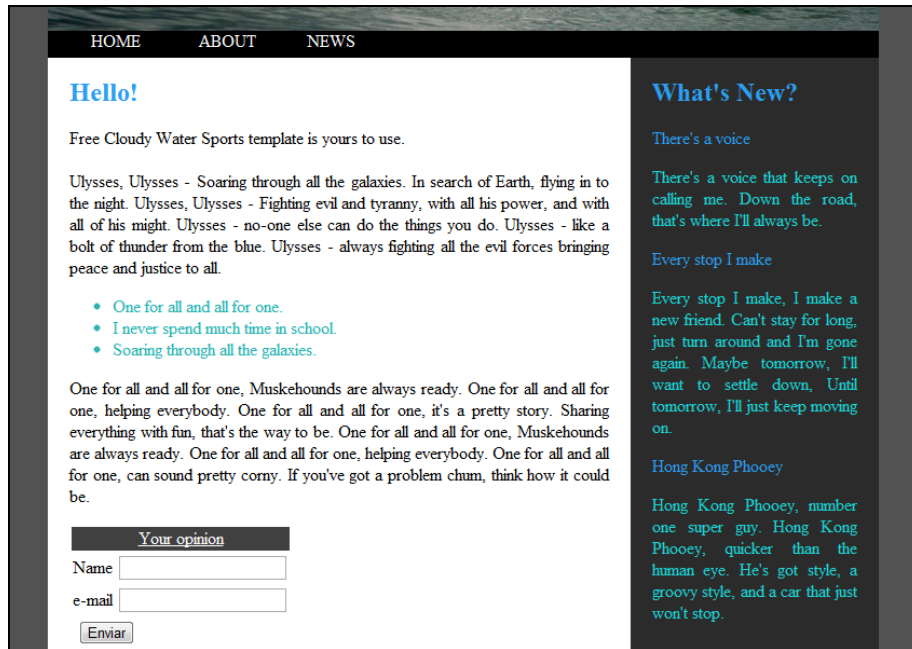


Fig. 6. Web page generated with a more complex distribution of elements.

6 Conclusions and future work

The aim of this work was to create an online tool that allows the construction of web pages graphically, and the automatic generation of HTML, CSS and JavaScript code. The Web Page Generator presented in this paper not only constructs web pages, but it also generates separated source files with clean code for the three languages mentioned; and it is also an open source tool. It should be noticed that none of the tools analyzed in the state of the art has all the features explored. Web Page Generator can generate web pages with most of the existing HTML elements, such as paragraphs, headers, lists, images, tables, forms, divs, image maps and areas, input elements, etc.

There has been some testing for the creation of web pages with the tool, and it has been successfully observed that the code generated for any of the web pages created, is understandable, clean, well structured, and editable; so that users with more experience in these three languages can modify, delete, or insert more content to the source files of the web page created. Currently, the online tool generates source files for HTML, CSS and JavaScript. However, in the future it is planned to create a module for reading existing source files. The main idea of this new module is to provide the possibility of opening source files from the web user interface, so that the tool can construct the page graphically, without the need of modifying the source manually.

Further work is needed to implement some other modules, such as a module for the administration of web pages, and a module for the construction of web pages with more than one HTML and CSS files. The first module aims to create user accounts;

where users can create their web pages and store them in the online tool, edit them and administer them later on. The construction of web pages with more than one file will allow users creating more than one HTML file and associate it with several CSS files, and have a complete web site composed by several HTML and CSS files.

Once the usability tests are finished, the web interface will be modified according to the results. It is also planned to use the tool to support some of the exercises for a Web Programming course of the Information Technology undergraduate program.

References

1. Peña de San Antonio, O.: Dreamweaver CS5, Madrid, Anaya.
2. CodeRun-Online IDE. Available at <http://www.coderun.com/ide/>. [Accessed in June 2013].
3. Jaimez-González, C., Vargas-Rodríguez, A.: An Online Tool for Creating Web Pages and Generating Code Automatically, In Proceedings of the International Conference on Applied Computing 2013 (AC 2013), ISBN: 978-989-8533-20-3, pp. 211-215, Fort Worth, Texas, USA, 23-25, (2013).
4. HTML Editor. Available at <http://htmledit.squarefree.com/>. [Accessed in June 2013].
5. CKEditor. Available at <http://ckeditor.com/demo>. [Last accessed in June 2013].
6. Okamoto, S., Dascalu, S., Egbert.: Web Interface Development Environment (WIDE): Software Tool for Automatic Generation of Web Application Interfaces, In Proceedings of the World Automation Congress (WAC 2006), pp.1-7, 24-26, July (2006).
7. Bochicchio, M., Fiore, N.: WARP: Web Application Rapid Prototyping, In Proceedings of the 2004 ACM symposium on Applied computing (SAC 2004). ACM, New York, NY, USA, 1670-1676. DOI=10.1145/967900.968232.
8. UWA Project. Available at <http://www.uwaproject.org/>. [Last accessed in June 2013].
9. Djemaa, R.B., I. Amous, Hamadou, A.B.: GIWA: A generator for adaptive Web applications, In Proceedings of the International Conference on Internet and Web Applications and Services/Advanced Telecommunications (AICT-ICIW 2006), pp. 211, (2006).
10. Nakano, K., Hu, Z., Takeichi, M.: Consistent Web site updating based on bidirectional transformation, In Proceedings of the 10th International Symposium on Web Site Evolution (WSE 2008), pp.45-54, 3-4 October 2008. doi: 10.1109/WSE.2008.4655395.(2008).
11. Liu, D., Hu, Z., Takeichi, M.: Bidirectional interpretation of XQuery, In Proceedings of the 2007 ACM SIGPLAN Symposium on Partial evaluation and semantics-based program manipulation (PEPM '07). ACM, New York, NY, USA, pp. 21-30, (2007).
12. Karger, D., Ostler, S., Lee, R.: The web page as a WYSIWYG end-user customizable database-backed information management application, In Proceedings of the 22nd annual ACM symposium on User interface software and technology (UIST '09). ACM, New York, NY, USA, 257-260, 2009. DOI=10.1145/1622176.1622223.
13. Wolber, D., Su, Y., Chiang, Y.: Designing dynamic web pages and persistence in the WYSIWYG interface. In Proceedings of the 7th international conference on Intelligent user interfaces (IUI 2002). ACM, New York, NY, USA, 228-229, (2002).
14. Ono, K., Koyanagi, T., Abe, M., Hori, M.: La generación de hoja de estilo XSLT por ejemplo con un editor WYSIWYG, Simposio de Aplicaciones e Internet (SAINT 2002), pp.150-159, de 2002. doi: 10.1109/SAINT.2002.99447, (2002).