BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2026
http://excel.fit.vutbr.cz

An interactive visual editor for teaching web design

Karolína Virágová

Abstract

This thesis focuses on improving the understanding of the structure of HTML documents during website development. It proposes and implements a tool that visualizes HTML elements in the form of a tree structure and integrates this representation with an HTML and CSS code editor. The system provides immediate visual feedback during code editing and enables a clear display of parent–child relationships between individual elements. The result is a tool that facilitates navigation within the code, accelerates the development process, and promotes a better understanding of the hierarchy of web documents.

xvirag00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Learning web design is often challenging for beginners, particularly because of the need to understand the structure of HTML documents and their syntax. The traditional approach, which relies on manually writing code, can be confusing and demotivating for students.

The goal of this thesis is to design an interactive visual editor that allows users to create websites intuitively without requiring immediate knowledge of HTML syntax.

Existing tools often either hide the code completely or, conversely, do not provide sufficient visual support. The proposed solution seeks to bridge these approaches.

The main contribution of this work is the creation of a tool that combines visual editing with the direct generation and display of HTML and CSS code.

2. Concept of a Visual Editor

This project was inspired by tools such as Scratch [1], which allow users to learn programming through a visual interface, and the W3Schools website [2], which provides interactive tutorials for web development. Both platforms simplify the learning process—Scratch through visual and playful means, and W3Schools through practical code testing.

Existing editors either hide the code or do not provide visual support for HTML and CSS

The proposed solution combines these approaches: it allows for intuitive visual editing of web page elements while simultaneously displaying the generated code. The editor thus provides immediate feedback, promoting a better understanding of HTML structure.

3. Architecture of visual editor

The architecture of the proposed system is divided into three main layers: the presentation layer, the application layer, and the data layer, as shown in Figure 1.

The presentation layer represents the interface through which the user interacts with the system. It consists of several components, including the Monaco code editor for writing HTML and CSS code, a visual editor for manipulating elements in a graphical form, a live preview of the resulting web page, and a panel for selecting predefined scenarios. The user performs actions such as editing code or modifying elements visually, which are then propagated to the underlying layers.

The application layer is responsible for processing user input and maintaining the internal representation of the document. It includes a tree structure that represents the HTML document hierarchy and a module for loading elements and their properties. This layer ensures synchronization between the visual representation and the source code.

The data layer provides the necessary data structures for the application. It contains a predefined dataset of HTML elements, their attributes, and allowed nesting rules, as well as predefined scenarios that can be loaded into the editor. These data are stored in JSON format and are used for validation and generation of

the user interface.

The interaction between layers ensures that any change made by the user is immediately reflected both in the visual editor and in the generated code, providing continuous feedback and supporting the learning process.

4. Generating HTML Elements and Attributes

A script was created to generate the structure of HTML elements; it uses several external libraries that provide a list of HTML tags, their attributes, and nesting rules. The result is a data structure that contains, for each element, its possible descendants
and attributes, including their values.

The script first compiles a list of all HTML elements. It then verifies the validity of the nesting for each element using the `isValidHTMLNesting` function and compiles a list of allowed descendants.

Next, the attributes are assigned to each element. If an attribute has predefined values, these values are included in the resulting structure. The output is an object representing the HTML element along with its properties, which can be further used, for example, for validation or generating a user interface.

The resulting data is stored in JSON format. The structure of the elements and their attributes is shown in Figure 2, where it is evident that each element contains a list of allowed children and attributes, with some attributes additionally having predefined values.

5. Displaying the HTML Structure

In the visual editor, an HTML document is represented as a tree structure corresponding to its hierarchical organization Figure 3. To implement this representation, the `Tree` component from the `react-arborist` library was used, which provides support for working with large tree structures, including
expanding and collapsing nodes or moving them
via drag-and-drop. Each node in the tree represents a single HTML element.

6. Conclusions

All of the main objectives of the project were successfully achieved. A web application was designed and implemented that allows for the visualization of an HTML document in the form of a tree structure and integrates it with an HTML and CSS code editor. The application provides immediate visual feedback and helps users better understand the relationships between individual elements.

Acknowledgements

I would like to thank my thesis advisor, Ing. Jiří Hynek, Ph.D., for his expert guidance, valuable advice, and support during the preparation of this bachelor's thesis.

References

[1] Mitchel Resnick, Natalie Rusk, John Maloney, Brian Silverman, Evelyn Eastmond, Karen Brennan, and Andrés Monroy-Hernández. Scratch – programming for everyone. https://scratch.mit.edu/, 2026. přístup 24. března 2026.

[2] W3schools online web tutorials. https://www.w3schools.com/, 2026. přístup 24. března 2026.