



Bilkent University  
Department of Computer Engineering  
Senior Design Project

**T2329**  
**Block Script**

**Project Specification Document**

Mehmet Kaan Örnek, 21901637  
Göksu Şenerkek, 21803341  
Khashayar Amini, 21903613  
Muhammed İkbāl Doğan, 21702990

Supervisor: Can Alkan  
Course Instructors: Atakan Erdem, Mert Bıçakçı

Nov 17, 2023

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

# Contents

<b>1 Introduction</b>	<b>3</b>
1.1 Description	3
1.2 High Level System Architecture & Components of Proposed Solution	4
1.3 Constraints	4
1.3.1 Implementation Constraints	4
1.3.2 Economic Constraints	4
1.3.3 Ethical Constraints	4
1.3.4 Language Constraints	5
1.4 Professional and Ethical Issues	5
<b>2 Design Requirements</b>	<b>5</b>
2.1 Functional Requirements	5
2.1.1 Design Tool	5
2.1.2 Compiler	6
2.2 Non-Functional Requirements	6
2.2.1 Usability	6
2.2.2 Reliability	6
2.2.3 Performance	6
2.2.4 Supportability	6
2.2.5 Extensibility	7
2.2.6 Scalability	7
<b>3 Feasibility Discussions</b>	<b>7</b>
3.1 Market & Competitive Analysis	7
3.2 Academic Analysis	7
<b>4 References</b>	<b>9</b>

# 1 Introduction

In today's technology-driven world, the importance of software development is recognized by everyone. Technology has been one of the most promising sectors for many years, therefore, there is a tendency among people towards learning programming. People, who have been working in other sectors, quit their job to learn programming and find a job in this sector. The computer science departments of the universities are one of the most popular choices among the most successful students in the high schools. In addition to individual tendency, there is a great effort to embed programming as a core skill to society, especially to children. For example, in Turkey, private high schools have started to offer programming courses [1]. Many countries, like the United Kingdom, India and South Korea, integrated programming into their national curriculum of either primary schools or high schools [2, 3].

However, teaching programming to children is a challenge by itself considering the abstract aspect of programming languages and the poor attention span of children. Therefore, the need for additional tools, that eases teaching how to program to kids by providing representational units against abstractness and gamification or appealing user interface against poor attention span, has emerged. It is observed that such a tool for web programming is absent, and this project aims to provide a tool for this purpose.

## 1.1 Description

Block Script will be a visual programming language designed for education purposes. Block Script will be used to make the process of learning web development easy and fun for children using blocks that encapsulates JavaScript, HTML and CSS codes. Using our design tool, users will be able to create projects and web pages within the projects. The needed blocks will be seen in the screen to create the desired pages. Once the user finishes the development, they will be able to export the underlying code that they have generated using blocks. Block Script will prepare the corresponding Javascript, HTML and CSS files, and deliver the files to the user. Therefore, Block Script will provide a comprehensive and educational web programming language by giving the user the ability to design both the user interface and client-side logic of a website without the need to write any line of code.

## **1.2 High Level System Architecture & Components of Proposed Solution**

Block Script will consist of two different components which are a design tool and a compiler. The design tool will be the interface that users will be integrating with. Using the design tool, users will be able to create projects and web pages within the projects by using and combining blocks. The needed blocks will be seen in the screen to create the desired pages. Once the user finishes the development, they will be able to export the underlying code that they have generated using blocks. To provide that code, we will have our compiler component. Compiler will generate the code and prepare the corresponding Javascript, HTML and CSS files which can be run as a website.

There will be mutual interaction between the design tool and the compiler. When a user wants to export their project, the information about which blocks are used and how they are combined with each other on which pages will be transmitted to the compiler component. The compiler will process this information and prepare the corresponding files. Once these files are prepared, it will transmit the files to the design tool, so that the users can download their resulting codes.

## **1.3 Constraints**

### **1.3.1 Implementation Constraints**

Implementation of the design tool will be done through Electron.js. Implementation of the compiler will be done with a high performance programming language like C/C++ or Golang. Version control will be done through Git.

### **1.3.2 Economic Constraints**

The product will be free for everyone. We believe educational opportunities should be equally accessible by everyone. It is aimed to publish this project as an open-source non-profit product. In addition, publishing it as open-source will open the product to contributions from others, thus, this approach will enhance the product's capabilities.

### **1.3.3 Ethical Constraints**

As personal data protection law suggests, any user information and data will not be shared with 3rd party users or companies. Our services will be designed to protect users' data from any leak.

### **1.3.4 Language Constraints**

The language we will be using in our product will be English. The most common programming languages are in English, and it is expected that there won't be any change on this in the near future. Since this project aims to provide an education-driven product, we believe learning and practicing the terminology in English is a part of the education towards the children who are new to programming.

## **1.4 Professional and Ethical Issues**

Source code of the software will be open for public access. Therefore, all the background processes will be transparent for everyone. User's data will not be collected without their permission. User data such as personal information and user generated content will be kept privately. It will not be accessible by any third party libraries or companies.

# **2 Design Requirements**

## **2.1 Functional Requirements**

### **2.1.1 Design Tool**

- Design tool must allow users to create a project.
- Design tool must allow users to create pages in the project.
- Design tool must show the code blocks available.
- Design tool must allow users to put code blocks to the created pages.
- Design tool must allow users to connect blocks linearly.
- Design tool must allow users to put blocks in blocks.
- Design tool must allow users to export the actual code that they created using blocks..
- The user interface of the design tool should be easy to use.
- The user interface should be engaging and fun to use.
- Tools provided by Block Script should be intuitive to use.
- All blocks needed to create a functional website should be accessible to the user.
- Compatible blocks should have compatibility in their visual design.
- Incompatible blocks should have incompatibility in their visual design.

### **2.1.2 Compiler**

- The compiler should create code which is readable.
- The compiler should export the generated code as JavaScript, HTML, CSS files.
- The compiler should give an error if there exists a problem in the design.
- The compiler should be fast and reliable.

## **2.2 Non-Functional Requirements**

### **2.2.1 Usability**

The design tool will be used mostly by children, therefore, use of the tool shouldn't be complicated, instead, it must be easy to understand and use. The texts on the product, like menu or help page, must be prepared in consideration of children. In addition, to increase usability against poor attention span of children, the use of tools must be fun and the interface must be appealing and attractive.

### **2.2.2 Reliability**

The design tool should be reliable while saving and loading designs to and from disk. Any issue during this process will cause loss of user progress. The compiler should be reliable while compiling each design and generate correct and accurate output.

### **2.2.3 Performance**

The designer user interface should be responsive and have high performance to provide a better user experience. The part of the product that will use the most computation time is the compiler. Therefore, the algorithms in the compiler must be optimized carefully to generate the output files in a short amount of time.

### **2.2.4 Supportability**

The design tool is supportable by Windows, Linux and web. It will be builded from the same codebase for any type of system. In terms of program life cycle, users' design state will be saved locally. States will be reachable when the user opens it next time. In terms of any exception, software should respond to those exceptions.

### **2.2.5 Extensibility**

The blocks in the design tool will be predetermined and listed in the design tool. However, as technology improves, the need for different kinds of blocks might emerge. Therefore, extensibility must be considered in the architecture.

### **2.2.6 Scalability**

The design tool and compiler should work cooperatively independent from numbers of users. As the user's computer has enough resources, it should handle the design and compilation process.

## **3 Feasibility Discussions**

### **3.1 Market & Competitive Analysis**

In the market for learning products, Block Script faces out against a number of other significant providers which are Scratch, Tynker and Code.org.

Scratch developed by MIT is one of the first visual programming languages that target kids. It discusses basic coding concepts and is popular among many schools across the world. Its main drawback in relation to Block Script is that it does not cover the web application languages such as JavaScript, HTML, or CSS. Just like scratch, another popular tool known as Tynker concentrates on basic programming. Nevertheless, it has a more narrow focus towards web development than Block Script does. Code.org has many courses but it targets the wider population compared to Block Script which involves the block-based interactive approach. All of them mainly focus on game design, animation or robotics.

Block Script's goal of building a website for and about kids makes it stand out from the competition. Many available marketing options do not extensively integrate JavaScript, HTML, and CSS on a child-safe block based interface at present. Thus, this puts the Block Script in one of a kind position since it fills a void that exists in Web development among young students.

### **3.2 Academic Analysis**

The academic validation of Block Script centers around two key aspects: block-based programming languages in education and educational necessity towards web development competencies in primary school students.

Many scholars have argued that the use of block-based programming languages fosters easier learning and especially, is very advantageous for kids. Through these tools, students are now able to understand some abstract programming concepts more easily with the use of tangibles. This approach for web development suits this study finding and Block Script.

As digital literacy becomes more important, web development should also be introduced at an early stage of education. The trend in academic affairs shows that more specific information technology courses like web development will soon be incorporated into elementary and high school syllabi [2-3]. This brings to mind the current fashion, which Block Script addresses in a direct manner, providing these skills within the scope of the child.

In order to strengthen and prove the Block script's approach, field studies should be undertaken to analyze its efficiency and superiority over other learning methods. For example, this process can include pilot programs in education institutions and assessment of learning outcomes.

Finally, Block Script seems to be on a good track for its target market — focusing on web development for children that many products already do not cover. It is also educationally relevant in the sense that it follows current trends and research on the usefulness of block-based programming languages. It is likely that further research would strengthen its academic validity and efficacy in teaching child-oriented web development.



## 4 References

[1] K. Arslan and M. Akçelik, "Programlama Eğitiminde scratch'in kullanılması: öğretmen Adaylarının Tutum ve algıları," National Education Academia Journal, <https://doi.org/10.32960/uead.455502> (accessed Nov. 15, 2023).

[2] C. Office, "D5 london: Teaching children to code," GOV.UK, <https://www.gov.uk/government/publications/d5-london-summit-themes/d5-london-teaching-children-to-code> (accessed Nov. 15, 2023).

[3] "New education policy 2020: Integration of coding and analytical thinking from the schooling level," Hindustan Times, <https://www.hindustantimes.com/education/new-education-policy-2020-integration-of-coding-and-analytical-thinking-from-the-schooling-level/story-G0ihYK8LitcMXmxbwB Bd6L.html> (accessed Nov. 15, 2023).