**Table of Contents**

[**Abstract** 2](#_Toc192626181)

[**Review of Literature** 4](#_Toc192626182)

[The Impact of ScratchJr on Young People 4](#_Toc192626183)

[Benefits of Learning a Programming Language 4](#_Toc192626184)

[Design of the platform 4](#_Toc192626185)

[**Review of Three similar Website** 5](#_Toc192626186)

[1. mBlock 5](#_Toc192626187)

[1.1 Introduction 5](#_Toc192626188)

[1.2 Features and Functionality 5](#_Toc192626189)

[1.3 Advantages 5](#_Toc192626190)

[1.4 Disadvantages 6](#_Toc192626191)

[2. Tynker 6](#_Toc192626192)

[2.1 Introduction 6](#_Toc192626193)

[2.2 Advantages 6](#_Toc192626194)

[2.3 Disadvantages 6](#_Toc192626195)

[3. KidsCode 7](#_Toc192626196)

[3.1 Introduction 7](#_Toc192626197)

[3.2 Advantages 7](#_Toc192626198)

[3.3 Disadvantages 7](#_Toc192626199)

[**References** 7](#_Toc192626200)

**Abstract**

Programming education has become an area of ​​growing concern today. Traditional programming languages ​​are too complex and unattractive for young beginners. But ScratchJr uses intuitive images instead of text for programming, which undoubtedly greatly reduces the difficulty of programming. This study explores how to design an educational platform specifically for young learners using ScratchJr as the main teaching material. The study will be conducted by developing an online learning platform designed to help young people take their first steps in programming. The goal is to increase the accessibility of coding, foster interest in programming, and develop a programming mindset.

ScratchJr is an entry-level programming language. It makes programming easy for young people to experience. ScratchJr provides a series of specific modules, such as: creating characters and backgrounds, moving left and right, jumping, zooming in, zooming out, rotating and some basic judgment conditions. These simple modules allow users to create a story or game. ScratchJr is simpler and more intuitive than Scratch. This is because ScratchJr is designed to use intuitive images for programming, rather than using language for programming like Scratch.

In today's era, learning coding has more advantages than disadvantages. But for young people, it is too difficult to directly learn to use programming languages ​​such as C, C++, Python, HTML, etc., and these programming languages ​​are more boring than ScratchJr and are not suitable as entry-level programming languages ​​for young people. Under these circumstances, we decided to create an educational platform for young people and only use ScratchJr as a teaching material to cultivate their programming thinking and interest.

The website design of an educational platform is crucial to the learning experience and stickiness of young users. Friendly interface design not only improves the convenience of operation but also stimulates users' interest in learning and participation. Effective design usually includes a good user interface and an interactive method that conforms to user habits.

For this reason, we wrote this literature review aims to explore how to design a ScratchJr teaching platform for young person. This study will create an online education platform to help young people take the first step to understand the world of programming. In this generation, full of electronic information, help young people develop programming thinking, creativity and the ability to overcome difficulties.

**Review of Literature**

The Impact of ScratchJr on Young People

ScratchJr is a very flexible tool with a relatively low barrier to entry. It works well for younger students with a relatively small set of instructions and the ability to create highly entertaining games with only a few blocks. It supports a wide range of activities, including games. In addition, it teaches students how to express themselves digitally and get excited about their creations, which was not easy to achieve before ScratchJr, as seen in conferences by Aye Thuzar and Aung Nay (2015).

Benefits of Learning a Programming Language

Learning programming is not only a process of mastering technical skills, but also an important way to develop problem-solving skills and logical thinking. Programming encourages students to acquire new skills that can help them solve various real-world problems. Students develop the ability to overcome difficulties, not give up, and achieve goals. For example, during the coding process, when testing a program, students may get wrong results, and they will not be able to move forward; but they can go back and try again. This method of repeatedly testing the program until it succeeds is very helpful for them and they will be successful in their future lives. In addition, learning a new language helps students better understand the world around them. The same is true for learning coding. Each character we can type on the keyboard has a special representation, which is 0 and 1. These 0s and 1s provide students with ideas to understand how technology works and functions. When students learn to code, it is a similar process to learning a new language. This can improve students' communication and logical thinking skills, as seen in the research of Shorena Abesadze and David Nozadze (2020).

Design of the platform

The interface of a platform is not just a visual display, but also a bridge for users to interact with the platform. For most young people, good interface design should have bright colours, which will be more favoured, as seen in the study of Andrew Large, Jamshid Beheshti, Valerie Nesset and Leanne Bowler (2007).

**Review of Three similar Website**

1. mBlock

1.1 Introduction

mBlock is a programming education software developed by Makeblock, primarily designed for youth coding education. It is based on Scratch 3.0 and supports both block-based coding and Python, enabling students to transition from visual programming to text-based coding. Additionally, it is compatible with Arduino, micro: bit, and Makeblock robotics kits, making it a versatile tool for learning. Widely used in STEAM education, mBlock helps students develop skills in coding, artificial intelligence and the Internet of Things.

1.2 Features and Functionality

mBlock provides a range of features suitable for learners of different ages and skill levels. It offers block-based coding, which allows beginners to programmed by dragging and dropping code blocks. For more advanced learners, it supports Python programming, helping students transition from visual programming to text-based coding. The software also includes artificial intelligence features such as speech recognition, image recognition and machine learning, making it a useful tool for STEAM education. Additionally, it supports hardware programming and is compatible with micro: bit, Arduino and Makeblock robotics kits, allowing students to engage in hands-on learning. The inclusion of IoT programming enables cloud-based networking, giving students the opportunity to explore IoT application development. mBlock is also cross-platform, with support for Windows, Mac, iOS, Android and web-based programming.

1.3 Advantages

Priambodo, Nugroho, and Hadiono (2023) found that the user interface design of mBlock has a significant positive impact on students’ interest and motivation in learning robotics. The study highlights that visual appeal, and clear instructions play a crucial role in enhancing the learning experience and increasing engagement. Additionally, mBlock integrates seamlessly with STEAM education by supporting various educational technologies that improve students’ logical thinking and problem-solving abilities. The study also suggests that the software’s block-based coding structure lowers the barrier to entry for beginners and makes programming more accessible. Güleryüz (2022) further emphasized that participation in mBlock-based robotic coding training significantly improves students’ confidence in coding, reduces anxiety related to programming, and enhances long-term retention of computational knowledge. Their study concluded that students who engaged in mBlock activities showed greater self-efficacy and a more positive attitude towards coding. The research also demonstrated that mBlock combination of drag-and-drop coding and hardware programming creates an interactive learning experience that encourages students to experiment, solve problems and develop computational thinking skills

1.4 Disadvantages

Despite its many advantages, Priambodo, Nugroho, and Hadiono (2023) found that the impact of an easy-to-understand user interface on students’ learning motivation was relatively weak. This suggests that while mBlock interface is generally well-received, some students may still require additional guidance and instructional support. The study also noted that there may be differences in how male and female students engage with the software, indicating that further refinements to its design could improve inclusivity. Güleryüz (2022) pointed out that although mBlock lowers the entry barrier for beginners, it may not fully meet the needs of more advanced users looking to work on complex projects. Their research indicated that students interested in progressing to higher-level programming languages such as Python or C++ might find mBlock capabilities somewhat limiting. Additionally, mBlock compatibility is primarily focused on Makeblock own hardware, which could restrict students who wish to explore robotics programming with third-party hardware. The study suggested that expanding hardware compatibility could enhance the flexibility of the software and provide students with more opportunities for exploration.

1. Tynker

2.1 Introduction

Tynker is an educational programming platform founded in 2012 that aims to teach children and beginners how to program. It offers video tutorials and uses a block-based coding interface, similar to ScratchJr. In addition, Tynker offers basic courses in Python, JavaScript, HTML, and CSS. Tynker offers a variety of courses, activities, and projects covering different aspects of programming, including game design, robotics, web development, and more. One of Tynker's accolades is that it is used in more than 150,000 schools around the world.

2.2 Advantages

In the study, Wafaa Elsawash and Rawy Thabet stated that the Tynker learning platform, with its interactive tools, fun activities, and puzzles, is an excellent solution for teaching programming to early learners. Tynker mainly uses videos as teaching materials, interspersed with visual codes to introduce logical concepts to children. During the video, an interactive environment is provided so that students can immediately try the code demonstrated in the video. According to the study, the analysis of activities and test results confirmed the success of the Tynker platform in teaching coding skills to young students. In addition, students' coding performance can be improved by learning and practicing in a fun and interactive way.

2.3 Disadvantages

Despite the many benefits offered by Tynker, Wafaa Elsawash and Rawy Thabet found that some authors questioned the practicality of block-based coding for learning actual code; they believed that when students turned to text-based programming, they would feel overwhelmed by the structure of textual programming languages. In addition, Wafaa Elsawash and Rawy Thabet also stated that since Tynker only teaches in English, this more or less hindered some students from fully understanding some programming concepts. The language barrier had a negative impact on students' academic performance.

**References**

1. <https://ieeexplore.ieee.org/document/7369023>
2. <https://www.ijlt.org/uploadfile/2020/0903/20200903031346728.pdf>
3. <https://asistdl.onlinelibrary.wiley.com/doi/full/10.1002/meet.1450430120>
4. <https://mblock.cc/>
5. <https://www.researchgate.net/profile/Hasan-Gueleryuez-2/publication/361851673_The_Effect_of_Robotic_Coding_Mblock_-Arduino_Activities_on_Students'_Self-efficacy_and_Attitudes/links/62c8320c00d0b4511040c7ff/The-Effect-of-Robotic-Coding-Mblock-Arduino-Activities-on-Students-Self-efficacy-and-Attitudes.pdf>
6. <https://www.researchgate.net/profile/Wisnu-Priambodo/publication/374029939_Impact_of_mBlock_Interface_Design_on_Student_Interest_and_Motivation_in_Primary_School_Robotics_A_Case_Study_in_Indonesia/links/650a940261f18040c20d74a7/Impact-of-mBlock-Interface-Design-on-Student-Interest-and-Motivation-in-Primary-School-Robotics-A-Case-Study-in-Indonesia.pdf>
7. <https://www.tynker.com/about/>
8. <https://www.researchgate.net/publication/364668329_The_Effectiveness_of_Tynker_Platform_in_Helping_Early_Ages_Students_to_Acquire_the_Coding_Skills_Necessary_for_21st_Century>