Enhancing Self-Efficacy in Computer Science Education

The Role of Large Language Models in Clarifying Error Messages for High School Students

Kris van Melis

Delft University of Technology

TU Delft

Enhancing Self-Efficacy in Computer Science Education

The Role of Large Language Models in Clarifying Error Messages for High School Students

by

Kris van Melis

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Tuesday June 17, 2025 at 14:00.

Student number:	5070619	
Project duration:	November 19, 2024 – June 17, 2025	
Thesis committee:	Prof. M.M. Specht	TU Delft, supervisor
	Prof. Dr. Mark Neerincx	TU Delft
	Dr.ir. E. Aivaloglou,	TU Delft, daily supervisor

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Contents

Summary

Computer Science education, particularly at the beginner level, often presents challenges due to vague and unhelpful error messages. This problem is particularly significant for students with low self-efficacy, leading to hindered learning experiences. Large Language Models (LLMs) offer a promising solution by generating more comprehensible and supportive error messages. This study aims to assess whether the rewriting of error messages using LLMs can improve self-efficacy among high school students, focusing on self-efficacy and study success as indicators of improved learning experiences. Through in-class experiments with 32 participants, the findings revealed that LLM rewritten error messages, although consistent with existing research, did not produce statistically significant effects. Therefore, more research is needed to evaluate their impact on learning outcomes and explore the most effective types of prompt. This research contributes to understanding the role of LLMs in educational settings, providing empirical insights into their effectiveness in real-world scenarios.

1

Introduction

In the field of Computer Science education, particularly for beginners, error messages often present a significant challenge due to their often vague and unhelpful nature [Campbell et al., 2014]. For example, syntax error messages frequently fail to pinpoint the exact location of an error, leading to confusion, or they might report errors located in unrelated lines [2014]. In addition, teachers might not always be available to help all students simultaneously [Mauriello et al., 1999]. This issue is especially pronounced for people who have low self-efficacy, as ineffective feedback can exacerbate feelings of inadequacy and hinder the learning process [Eltegani and Butgereit, 2015].

On the other hand, positive coding experiences can improve self-efficacy, particularly among female students [Phillips and Brooks, 2017]. Large Language Models (LLMs) provide a promising solution by potentially generating more comprehensible and supportive error messages [Leinonen et al., 2023]. Such improvements may improve the coding experiences for students, thus fostering a greater sense of competence and persistence [Beyer, 2014].

This research will specifically examine whether the use of Large Language Models (LLMs) to rewrite error messages can enhance self-efficacy among high school students. Our main academic contribution is the in-class testing of LLM-rewritten error messages with high school students, thereby providing empirical insights into their effectiveness in real-world educational settings. Furthermore, we will investigate any potential differences in the results between male and female students, since previous research [Phillips and Brooks, 2017] found that programs such as the hour of code significantly boosted self-efficacy among girls more than among boys.

In the Netherlands, HTML is often the first programming language introduced to students enrolled in Computer Science courses at institutions like Stanislascollege Westplantsoen or Emmauscollege. Consequently, we concentrate on HTML, since it represents students' initial experiences with text-based programming.

Research Question 1: How is the use of an LLM to rewrite HTML syntax error messages related to students' self-efficacy compared to conventional error messages?

Research Question 2: How does the integration of an LLM to rewrite HTML syntax error messages relate to study success compared to conventional error messages?

Research Question 3: Does gender moderate the relationship between intervention type and self-efficacy outcomes?

To address these questions, we conducted an experiment involving 67 students from 6 classes of Adelbert College Wassenaar. The students were randomly assigned to either a control group or a test

group. The control group received syntax error messages from Slowparse from Mozilla, while the test group received syntax error messages from GPT-4o.

Our study reveals that while the LLM-assisted feedback system did not demonstrate statistically significant improvements over traditional feedback methods, such as Slowparse, in self-efficacy or study success, the findings align with existing research [Santos and Becker, 2024]. Furthermore, we did not observe significant differences in self-efficacy between male and female participants. The primary reason might be that the platform used was already optimized for interactive learning, which makes the intervention insufficient to induce measurable change. Moreover, the control group's error messages were already designed to be more readable and beneficial. It is also crucial to note that our study did not evaluate whether the students ultimately acquired more knowledge; it is possible that one of the groups might have retained more information. More research is needed to explore whether LLMs can improve the helpfulness of error messages.

2 Background

Before exploring the potential contributions of Large Language Models (LLMs) in addressing educational challenges, it is essential to first grasp the concept of self-efficacy and understand the processes involved in its formation.

2.1. Self-Efficacy in Learning Contexts

Self-efficacy, a concept pioneered by Bandura, defines an individual's confidence in their ability to successfully organize and implement strategies needed to handle anticipated situations [Bandura, 1982]. This belief is pivotal in determining how individuals approach goals, tasks, and challenges, with higher levels of self-efficacy often linked to increased motivation and persistence in learning environments [Zimmerman, 1995]. Bandura outlined four main sources that contribute to self-efficacy: mastery experiences, vicarious experiences, verbal persuasion, and physiological feedback [Bandura, 1977]. Among these, mastery experiences, which encompass personal achievements and setbacks, are viewed as the most crucial factor in forming beliefs about self-efficacy [Kleppang et al., 2023]. In educational environments, these successful experiences are crucial, as they reinforce the confidence of students in their abilities, directly contributing to improved academic performance and greater persistence [Schunk, 1987]. Understanding and nurturing self-efficacy is therefore essential for fostering not only individual academic success but also broader educational equity, as it influences a student's ability to overcome challenges and persevere through adversity.

2.2. The Importance of Positive Experiences in Computer Science Education

Positive experiences in computer science are essential to cultivate self-efficacy [Beyer, 2014]. In particular, coding experiences significantly influence self-efficacy compared to other computer-related activities [Hasan, 2003]. Initiatives like the Hour of Code are specifically designed to foster positive coding experiences, addressing disparities in engagement and achievement. These programs offer activities that are proven to improve self-efficacy, particularly among female students, by providing them with encouraging and affirming experiences in coding [Phillips and Brooks, 2017].

In contrast, failure can significantly decrease self-efficacy related to the task [Smith et al., 2006]. Therefore, it is crucial to minimize negative experiences during coding activities.

2.3. Challenges with Syntax Errors in Coding Education

Syntax errors coupled with vague error messages are a frequent source of frustration and a significant challenge, often leading to negative coding experiences among novice programmers [Campbell et al., 2014]. So, one simple error of a missing bracket can produce many other errors in the same file.

When students do not receive clear and timely feedback on their mistakes, it can result in feelings of

frustration, which hinder the learning process [Eltegani and Butgereit, 2015].

2.4. Common Obstacles in Learning HTML

Learning HTML frequently poses challenges for students, especially in the area of coding development. A study investigating help-seeking behaviors within a web development course revealed that the majority of help-seeking instances were specifically related to the development process [Park and Wiedenbeck, 2011].

Many students face significant difficulties in identifying errors in their code and, as noted in [Mauriello et al., 1999], teachers often struggle to assist all students simultaneously. In this context, effective error messages become essential as they can help reduce the teacher's workload and enable students to utilize their time more efficiently.

2.5. Large Language Models as a Solution

Qualitative analysis suggests that large language models (LLMs) have the potential to address these challenges by generating more helpful error messages. The research carried out by [Leinonen et al., 2023] indicates that expert evaluations of LLM error messages found them to be more helpful compared to traditional error messages. Furthermore, a quantitative study involving 8,762 online students from 146 countries, conducted by [Wang et al., 2024], demonstrates that students receiving error messages generated by OpenAI's GPT repeated errors 23.1% less frequently in subsequent attempts and resolved errors in 34.8% fewer additional attempts compared to those using standard error messages. Thus, by improving the clarity of error messages and helping students solve errors, LLMs could contribute to more positive coding experiences.

However, a qualitative study by [Santos and Becker, 2024] involving 116 university students reported that handwritten error messages outperformed GPT-generated error messages when dealing with buggy C programs. Despite this finding, there has been limited in-class research on this topic. According to [Leinonen et al., 2023], more research is necessary in the classroom. High school represents a critical period for the development of self-efficacy beliefs [Bandura, 1997], yet to our knowledge, no quantitative research has been conducted at the high school level on how LLMs can rewrite error messages.

3

Methodology

3.1. Research Design

This study used a randomized test and control group pre-posttest approach to examine the impact of Large Language Model (LLM)-assisted feedback on student motivation and self-efficacy in learning HTML. The research has been conducted in a high school setting, involving students 16 years and older. Students aged 16 years and older were chosen because they can provide consent for the research themselves. Ethics approval was obtained from the Human Research Ethics committee of the TU Delft prior to the start of the experiment. Participants are randomly assigned to one of two groups: an test group receiving LLM-assisted feedback and a control group receiving HTML syntax error messages from Slowparse.

Before starting the experiment, an in-class teacher provides students with a brief introduction to HTML and a comprehensive overview of the experiment. This introduction aims to give them a solid understanding of HTML, enhancing their ability to respond accurately to the pre-survey questions. The platform includes an assistant named Chippy, designed to assist with these exercises.

Students are instructed to first seek help from Chippy and, if they still need assistance, they can reach out to a teacher. The introductory session lasts about 10 minutes, followed by a 30-minute period during which students complete the prior knowledge survey, complete the pre-self-efficacy survey, and work on the exercises. After this period, students are automatically directed to the post-self-efficacy survey to complete the data collection.

The exercises begin with a brief introduction to HTML, after which the students proceed to create webpages by independently writing tags such as `<h1>`, `<p>`, `<b>`, `<ul>`, `<li>`, and others. These exercises have previously been used in high school classes using the Bitsized learning platform. To better evaluate the impact of the intervention, additional exercises have been incorporated that specifically require students to identify and correct syntax errors. This ensures that even students who are relatively proficient at HTML will encounter syntax issues, thereby making the intervention's effects more noticeable.

3.2. In-class explanation

The in-class session begins with the teacher introducing themselves and sharing their research focus, which involves exploring the intersection of large language models in the field of education. The teacher proceeds to explain that the day's lesson will cover the creation of webpages, with the assistance of an AI tool named Chippy. This introduction is followed by a brief explanation of HTML, which stands for Hypertext Markup Language. The teacher articulates that "hypertext" refers to non-linear navigation, while "markup language" involves the annotation of text, akin to styling text in a word processor to highlight it. The students are then provided with an example of a paragraph tag, including its opening tag, content, and closing tag, as well as an illustration of how text can be marked within a paragraph tag. Following this, a basic overview of a webpage's structure is given, emphasizing the doctype and the html tag's head and body sections. It is noted that, for the time being, students should place all of

their code within the body tag.

Additionally, the teacher discusses the necessity of obtaining consent to collect and use the students' data for research purposes. The importance of providing truthful responses when giving consent is highlighted. Students are reassured that they can choose not to provide consent and still participate in the learning exercises without receiving a grade. Their data will remain anonymous, and their performance will not be graded. The potential benefits of the research in improving educational methods are underscored.

Next, the class rules are outlined: students are expected to work quietly and independently and to refrain from assisting each other. They are encouraged to give their best effort and to use Chippy, the AI, if they encounter difficulties. If further assistance is needed, the teacher is available for help. It is emphasized that students should carefully read the exercises, as they will not have the opportunity to revisit them.

Once these instructions are delivered, students are invited to ask any questions before being directed to access the test environment.

3.3. Research Environment

To assess the effectiveness of LLM support in helping students, we developed a new learning environment based on Bitsized[1]. Bitsized is a Dutch method for teaching Computer Science and has been implemented in several Dutch high schools. Our learning environment used Bitsized's interactive platform, which includes HTML exercises that can be solved with a live preview. The students' code is automatically checked for syntax and correctness, and if a syntax error occurs, an error message is provided by GPT-4 by default. While our environment retained most elements of Bitsized, we incorporated additional features such as surveys and action-tracking capabilities to measure the effects of the intervention and to facilitate the intervention itself.

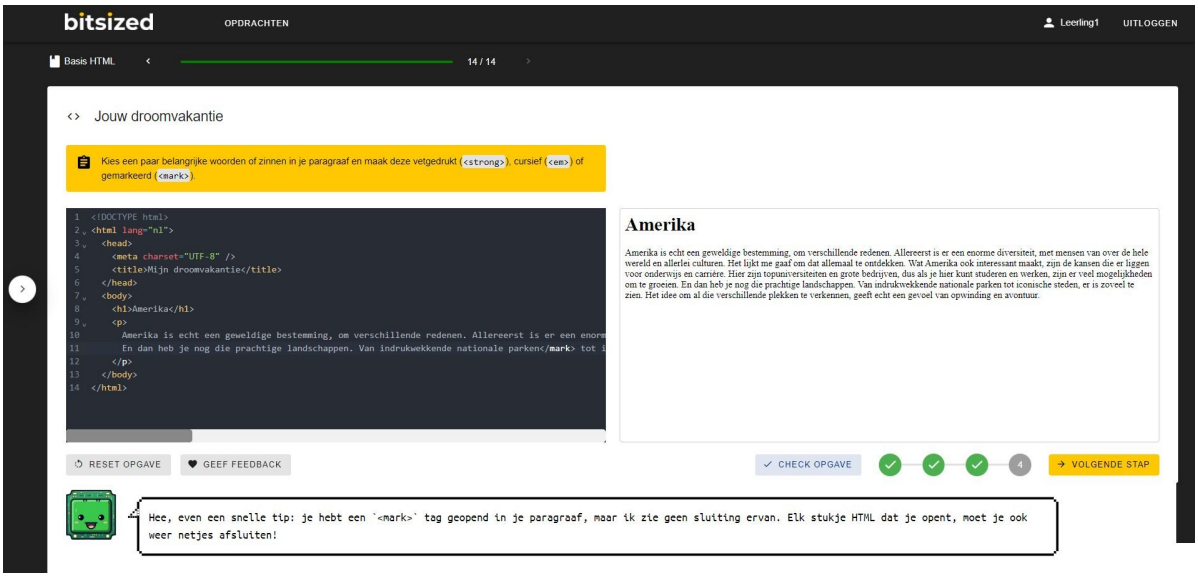


Figure 3.1: A screenshot of the platform Bitsized

3.4. HTML Exercises

To evaluate our intervention, participants engaged with HTML exercises in the learning environment, mainly sourced from Bitsized itself with some modifications. One significant change was the removal of a video lecture due to the impracticality of supplying headsets to all students; instead, this lecture's content was covered through in-class explanations. Additionally, a series of exercises were introduced that required participants to correct pre-existing syntax errors in the code. These additional exercises

[1] https://bitsized.nl

were designed to increase the likelihood of participants encountering syntax errors, thereby facilitating a more distinct comparison between the test and control groups in the intervention. An overview of all exercises and their content is presented in Table 3.1.

Activities of type ”survey” require participants to input data. Activities of type ”example” include an explanation and require participants to complete an exercise to proceed. Activities labeled as ”exercise” necessitate that participants complete an exercise to advance. Exercise 1 is presented before the pre-self-efficacy survey to reiterate the in-class explanation about HTML. This approach ensures that students have a clear understanding of HTML, enabling them to accurately respond to the self-efficacy survey. Upon completion of their allotted time for working on the exercises, participants were automatically redirected to the post self-efficacy survey.

Name	**Type**	**Goals**
Consent	Survey	Ask for consent
Demographics	Survey	Obtain demographics information
Prior Knowledge	Survey	Obtain information about prior experience coding
Exercise 1	Example	Explain Hypertext Explain Markup language Give example of HTML snippet
Pre Self-Efficacy survey	Survey	
Exercise 2	Example	Show example of HTML page with code editor
Exercise 3	Example	Show specific example of h1 tag, opening, content, closing
Exercise 4	Exercise	Create h1 tag Create p tag
Exercise 5	Exercise-repair syntax	Fix syntax error of unclosed h1 tag
Exercise 6	Exercise-repair syntax	Fix syntax error of misspelled h1 tag
Exercise 7	Example	Example of nesting strong, i and b tags in a paragraph
Exercise 8	Exercise	Create b tag Create br tag Create i tag Create mark tag Correctly nest all tags in p tag
Exercise 9	Exercise-repair syntax	Repair incorrectly closed b tag Repair unclosed i tag
Exercise 10	Exercise	Create unordered list
Exercise 11	Exercise-repair syntax	Fix incorrectly closed li tag Fix incorrectly closed ul tag
Exercise 12	Exercise	Create ordered list
Post Self-Efficacy survey	Survey	

Table 3.1: Ordered list of activities and exercises with corresponding goals.

3.5. Intervention

The focus of the intervention is on the feedback mechanism activated in response to syntax errors. Specifically, when a user seeks a hint or attempts to validate their exercise, an HTML validator is run first. If this validator detects errors, the LLM is activated using the prompt detailed in Section 3.6. If no errors in validation occur, the exercise is further examined using HTML selectors to ensure that all elements are correctly configured on the page. Should any elements be improperly placed, a specific pre-written hint is provided; otherwise, the exercise is marked as complete. Consequently, the intervention activates only when a syntax error occurs. For the purpose of collecting more data, additional exercises with built-in syntax errors have been included, encouraging students to encounter

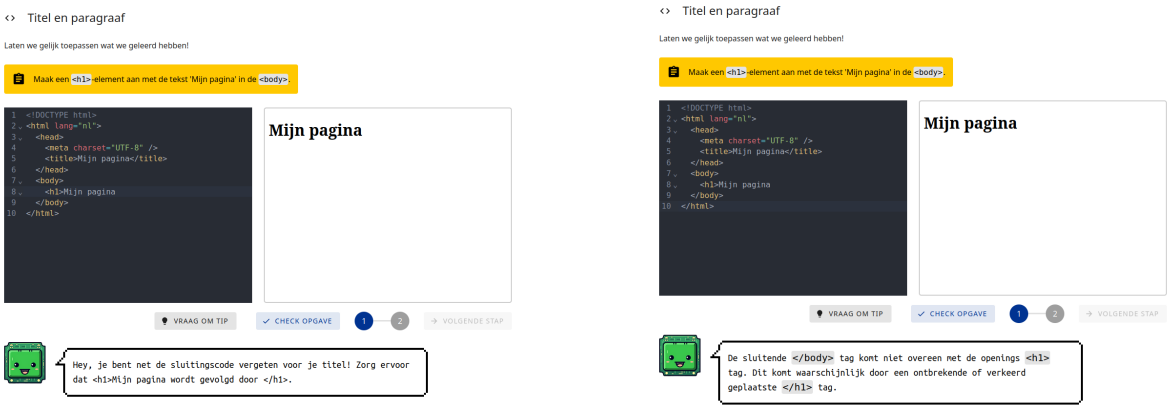
and benefit from the intervention more frequently, potentially leading to richer data outcomes.

Table 3.2: Comparison Between Test and Control Groups

Scenario: Unclosed Italic Tag `<i>test<i>`		
Aspect	**Test Group (GPT-4o)**	**Control Group (Slowparse)**
Error Message	Hey, check your `<i>` tags; the closing tag is missing a slash. Make sure the opening and closing tags match: `<i>text</i>`.	”The closing `</body>` tag here doesn’t pair with the opening `<i>` tag here. This is likely due to a missing or misplaced `</i>` tag.”
Guidance	Offers a hint to check tags and ensure pairing	Standard notifications without instructional feedback

3.5.1. Test Group

Students assigned to this group will receive feedback on their HTML assignments through a tool assisted by a Large Language Model. This tool is specifically designed to provide feedback only when a syntax error is detected in the student’s code. The LLM has been configured according to the prompt found in Section 3.6. The goal is for the LLM to provide feedback in a manner similar to that of a teacher.



(a) The test group interface

(b) The control group interface

Figure 3.2: Interface comparison between the test and control groups. Only the given error messages differs.

3.5.2. Control Group

Students in this group will rely on conventional HTML syntax error messages provided by Slowparse. Slowparse, an HTML parser developed by Mozilla, is designed to provide more user-friendly syntax error notifications by offering additional context on what is wrong and possible solutions [Mozilla, n.d.]. It is implemented in the ”Intro to HTML” module on Khan Academy, a major online learning platform [Khan, n.d.]. For example, when encountering an unclosed italic tag, such as `<i>test<i>`, a typical parser might display an error message like ”Unclosed element ’i’.” In contrast, Slowparse offers a more detailed message: ”The closing `</body>` tag here doesn’t pair with the opening `<i>` tag here. This is likely due to a missing or misplaced `</i>` tag.” Although these messages effectively highlight errors, they do not offer guidance or instructional feedback beyond standard notifications.

An example comparison of error messages from the test and control group can be found in Table 3.2.

3.6. LLM model & Prompt for HTML Syntax Error Feedback

This section details the model used and the design and operational framework of the prompt, specifically crafted to assist students by evaluating and providing constructive feedback on syntax errors in their HTML code. The prompt embodies the AI instructor, Chippy, who provides rewritten error messages. We opted to use the same GPT-4o model and prompt since these were already utilized within Bitsized, which has been tested and proven effective with multiple teachers and classes.

The prompt is designed by Bitsized to provide students with teacher-like support, offering concrete guidance and hints without revealing the corrected code. The prompt has been written, developed, and tested with GPT-4o model from OpenAI. Instructions have been implemented to ensure that Chippy clearly explains errors when the standard template is incorrectly modified, resulting in a syntax error. Furthermore, these instructions also help keep the error messages concise and ensure that they are correctly formatted for display in the platform.

3.6.1. Functionality and Design

- **Role of AI as a Teacher (System):** The AI adopts the persona of a teacher, Chippy, who supports students by identifying syntax errors in their HTML code. Rather than offering direct solutions, Chippy provides hints to cultivate an educational atmosphere where students learn by engaging with their mistakes.
- **Feedback Protocol:**
  - Feedback is communicated in an approachable and informal manner, making it suitable for high school students.
  - Hints are concise, restricted to two sentences, and limited to addressing one error per instance to allow students to concentrate on individual mistakes.
- **Code Validation Framework:** The AI assesses students' code against a standard HTML template to ensure that essential components are included and correctly implemented. The AI identifies and points out missing or incorrect elements within this framework that commonly lead to syntax errors directly to the student. Otherwise, it would often fail to point out the error message found by the HTML parser.
- **Student Code Submission (User):** The student submits a segment of HTML code for analysis, serving as input for the AI to evaluate and generate feedback on syntax errors.
- **Error Detection and Feedback Communication:**
  - Syntax errors identified in the student's code are communicated in a way that ensures clarity, with angle brackets (<, >) replaced by HTML-safe versions (`<`, `>`) for proper display on web interfaces.

3.6.2. Prompt Messages

The following prompt messages have been used:

1. **System Role:** The AI assumes the role of a teacher named Chippy, providing hints and pointing out errors without disclosing complete answers, using student-friendly language and handling errors one step at a time.

   > "You are a teacher named Chippy assisting a student with a task. Provide hints without giving answers, identify the incorrect code, and limit tips to two sentences. Tackle one mistake at a time and use language understandable to high school students, responding informally."

2. **System Role:** The AI verifies code against a standard template, flagging any omissions or deviations.

   > "The code is checked by a validator that flags errors for missing elements from the standard code. The standard template includes:
   >
   > ```
   > <!DOCTYPE html>
   > <html lang="nl">
   > ```

```
    <head>
        <meta charset="UTF-8" />
        <title>Mijn pagina</title>
    </head>
    <body>
    </body>
</html>
```

3. **User Role:** The code of the student gets provided for feedback.

   "My code is: `[students code]`"

4. **System Role:** The AI highlights syntax errors, formatting hints for HTML readability.

   "There are one or more syntax errors in the student's code. Can you explain the errors to the student? Format the hint to be HTML-readable, replacing < > with `< >`.

3.7. Pilot Test

To evaluate the experimental setup, a pilot test was conducted with a small class of 7 High School students from Adelbert College Wassenaar. The test revealed that the students needed further explanation regarding HTML, especially concerning fundamental concepts such as initiating HTML code. Many students struggled to understand where to begin typing HTML code within the body tag, which is intended to make content visible on the webpage. This challenge resulted in difficulties with the first exercise, leading many students to require teacher assistance to progress. In addition, a flaw was found in the exercise checker that is responsible for determining the correct completion of the exercises. As a result, students experienced difficulties with the faulty exercise, unable to continue with the correct code. This error has been corrected.

3.8. Hypotheses

RQ1: How does the use of an LLM to rewrite HTML syntax error messages relate to students' self-efficacy compared to conventional error messages?

- **H0:** There is no significant relationship between the use of a Large Language Model (LLM) for rewriting HTML syntax error messages and students' self-efficacy compared to conventional error messages.
- **H1:** There is a significant positive relationship between the use of a Large Language Model (LLM) to rewrite HTML syntax error messages and students' self-efficacy compared to conventional error messages.

RQ2: How does the integration of an LLM to rewrite HTML syntax error messages relate to study success compared to conventional error messages?

- **H0:** The integration of a Large Language Model (LLM) in rewriting HTML syntax error messages does not have a significant relationship with study success compared to conventional error messages.
- **H1:** The integration of a Large Language Model (LLM) in rewriting HTML syntax error messages has a significant positive relationship with study success compared to conventional error messages.

RQ3: Does gender moderate the relationship between intervention type and self-efficacy outcomes?

- **H0:** Gender does not moderate the relationship between the intervention type and self-efficacy outcomes.
- **H1:** Gender significantly moderates the relationship between the intervention type and self-efficacy outcomes.

3.9. Data Collection

3.9.1. Demographics

To analyze our participant base, the following information gets collected: age, gender, educational level. Gender is collected to see if there are differences between those who identify as male or female. The educational level and age are collected so that this experiment could be performed again and provide context for the results.

3.9.2. Prior Knowledge Survey

To prevent the dataset from being skewed by students with previous programming experience, a prior knowledge survey will be used to gauge their prior exposure. Students will respond using a 5-point Likert scale: *none, little, average, much, a lot*.

The first question asks: ”How much programming experience do you have? (e.g., with Scratch, Code.org, Python, Javascript, Java, C#, HTML/CSS)” to assess whether they have any coding experience.

If a student selects *average* or higher, a second question follows: ”How much experience do you have with text-based programming languages? (e.g., Python, Javascript, Java, C#, HTML/CSS)” to gauge their familiarity with text-based programming.

If the student once again selects *average* or higher, a third question is posed: ”How much experience do you have with debugging your code?” to evaluate their debugging skills.

This approach allows us to potentially exclude individuals with above-average experience and, with a sufficiently large sample size, to explore how their behavior may differ from students with little or no experience.

3.9.3. Pre- and Post-Study Self-Efficacy Surveys

All participants are required to complete a survey at the beginning and end of the study. These surveys aim to evaluate participants' self-perceived efficacy in learning HTML, employing a modified version of the Discipline Scale Self-Efficacy Survey originally developed at Imperial College London [“Self-efficacy in discipline scale”, n.d.].

Although there are Dutch self-efficacy surveys available, such as the one developed by Schwarzer, R. and Jerusalem, M. in 1995 [Schwarzer, 1995], and its Dutch adaptation by Teeuw, B., Schwarzer, R., and Jerusalem, M. in 1994 [Teeuw et al., 1994], these surveys are not specifically designed to assess self-efficacy in defined subjects, but rather general self-efficacy. Furthermore, these surveys might introduce response bias due to their use of agree-disagree response formats. Research by Imperial College has highlighted that such formats can lead participants to agree with statements irrespective of the content [“Self-efficacy in discipline scale”, n.d.]. This tendency is known as acquiescence bias, as discussed by Wright in 1975 [Wright, 1975]. Furthermore, this method places greater demands on participants, which can potentially decrease the quality of the data collected [Fowler Jr and Cosenza, 2009]. Saris et al. also noted similar concerns about the impact of survey design on data quality [Saris et al., 2010].

For this study, the selected survey has been adapted to Dutch, aligning with the translation used by vanMaanen in 2021 [van Maanen, 2021]. Subsequently, the survey content was specifically tailored to relate to the context of the HTML module. This adapted version of the survey can be reviewed in Table 3.3 and the Dutch version can be found in Appendix A.

3.9.4. Platform Analytics

Apart from collecting survey and demographic data, we will also record detailed analytics of how students interact with the platform and perform on the exercises. Platform analytics will be logged per type of action, each associated with a timestamp. The following platform analytics are collected:

Typing Time

Typing time are critical indicators of engagement and efficiency in the problem-solving process. The typing time refers to the period during which a student actively interacts with the exercises by typing. If there is no typing activity for a duration of 5 seconds, the system logs that the user has stopped typing.

Answer the following questions while thinking of yourself as a student taking this HTML course:

Question	Not at all confident	Slightly confident	Somewhat confident	Quite confident	Extremely confident
How confident are you that you can complete all the work that is assigned in your HTML module?					
When complicated ideas are presented in your HTML module, how confident are you that you can understand them?					
How confident are you that you can learn all of the material presented in your HTML module?					
How confident are you that you can do the hardest work that is assigned in your HTML module?					
How confident are you that you will remember what you have learned in your current HTML module next year?					

Table 3.3: Self-assessment questions for self efficacy in the HTML module

Prolonged idle times may suggest that students are taking time to think through the problem or are potentially encountering difficulties.

Exercise Status Logs

We will log when a user starts a new exercise and when a user completes an exercise. These logs will help to understand user engagement, identify the duration of activity engagement, and evaluate how far they progress through the exercises.

Exercise Hint and Solution Check Analysis

We will log interactions when a user checks an exercise by either asking for a hint or checking the exercise solution. This includes logging whether they pressed the ”ask hint” or ”check exercise” button, allowing us to analyze differences based on gender or test/control group classifications. This data will also show how many attempts a user needs to solve an exercise and whether tips help complete the exercise successfully.

Feedback and Error Logging by Assistant Chippy

Additionally, we will log the user’s current code and the type of error they made: syntax or non-syntax errors. For syntax errors, we will also document the feedback provided by the assistant, Chippy. This logging will enable us to assess the effectiveness of the feedback in aiding students’ problem-solving processes.

3.10. Data Filtering

To achieve accurate results, we filter the data by removing any records in which the participants did not fully complete both the pre- and post-self-efficacy surveys. This includes participants who were under 16 years of age and therefore were withdrawn before data collection occurred. Furthermore, participants who reported having average or above experience with programming (question one of the prior knowledge survey) are excluded, as they might represent outliers. Lastly, participants who have not completed at least two exercises are removed, as they likely encountered difficulties and can be considered outliers.

3.11. Participants

The study used high school students aged 16 and above from Adelbert College Wassenaar, encompassing a total of six classes. Initially, 67 records were considered for inclusion, but after applying

exclusion criteria, eliminating 20 participants under the age of 16 years and 5 participants with excessive prior knowledge, the pool of participants was refined to 32 usable records, representing 47.8% of the original sample. Each of these 32 students has provided their written consent via the form in Appendix B. To facilitate proper randomization, students will be randomly assigned to the test or control group through the designated platform. Measures will be taken to prevent cross-contamination between groups by instructing students not to engage in interactions that might lead to mixing between group members.

3.12. Data Analysis

3.12.1. Demographics

We will present the distribution of demographics, including age, gender (male, female, or none of the above), and educational level.

3.12.2. Syntax Error Analysis

We will examine the frequency and types of syntax errors most commonly made by students to gain a comprehensive overview of the data.

3.12.3. Markov Chain Analysis

To investigate how participants interacted with the learning environment, we will perform a Markov chain analysis of their behaviors. Instead of creating a separate Markov chain for each exercise, all exercises will be combined into a single comprehensive Markov chain to identify the most common paths taken by the participants. The chain begins with an initial node, representing the start of an exercise, and transitions may lead to the following nodes: Typing, Requesting a Hint, or Checking the Exercise. If a participant requests a hint, they may then transition to providing feedback on either a syntax error or another type of error. If a participant checks the exercise, potential transitions include encountering a syntax error, a non-syntax error, or completing the exercise, which represents the terminal node. Our analysis will also examine this unified Markov chain across different groups, such as control versus test groups and gender-based comparisons, to gain deeper insights into participants' interaction patterns.

3.12.4. RQ1: Self-Efficacy

To assess the impact of using a Large Language Model (LLM) to rewrite HTML syntax error messages on student self-efficacy, we conducted an Analysis of Covariance (ANCOVA). This approach allowed us to adjust for baseline differences in self-efficacy and isolate the effect of the intervention. The following details the components and methodology of our analysis:

Components of the ANCOVA Model

- **Independent Variable**: The intervention method, distinguishing between participants receiving error messages rewritten by an LLM and those receiving conventional error messages.
- **Dependent Variable**: Students' self-efficacy scores following the intervention, as measured by a standardized self-efficacy scale.
- **Covariate**: Self-efficacy scores prior to the intervention, serving to control for initial differences in self-efficacy across participants.

Methodology

Model Specification The ANCOVA model was specified to examine the relation between intervention type and post-intervention self-efficacy, while controlling for pre-intervention self-efficacy. This model enables us to determine the unique contribution of the intervention.

Assumptions and Validation We checked the assumptions of ANCOVA, including the normality of the residuals and homogeneity of the regression slopes, ensuring the validity of the model in capturing group differences. Any deviations from these assumptions were carefully evaluated and addressed.

Outcome Metrics The ANCOVA model provided estimates for:

- **Adjusted Group Means**: Representing the self-efficacy levels for each group after adjusting for pre-intervention scores.
- **Effect Sizes**: Measured using Partial Eta-Squared to quantify the magnitude of the intervention's impact.
- **Significance Testing**: Assessing whether the observed differences between the groups are statistically significant.

Interpretation and Implications

The results of the ANCOVA offer insights into the relationship between intervention methods and student self-efficacy:

- **Higher Adjusted Means in the LLM Group**: Suggests that rewriting error messages using an LLM positively impacts students' self-efficacy, enhancing their confidence in addressing HTML syntax errors.
- **Significant Effects**: Indicate that the intervention approach has a tangible effect on self-efficacy, warranting consideration in educational practice.

Sensitivity Analysis To confirm the robustness of our findings, sensitivity analyzes were performed. These analyzes assessed the stability of results against variations in model assumptions and potential covariates, ensuring that the conclusions remain consistent under different analytic conditions.

This comprehensive analysis enables a deeper understanding of how LLMs can be effectively integrated into educational tools to boost student self-efficacy and improve learning experiences.

3.12.5. RQ2: Study Success

To comprehensively evaluate study success, we constructed a composite index that integrates several key performance metrics. This index combines the success rate, syntax mistake frequency, and error correction efficiency into a single, standardized measure of overall performance. The following outlines the development and methodology of our composite index:

Components of the Composite Index

- **Success Rate**: Defined as the percentage of successful attempts. A higher success rate indicates greater proficiency and understanding.
- **Syntax Mistake Frequency**: Measures the total number of syntax mistakes per exercise solved. Fewer mistakes suggest better grasp of the material and application skills.
- **Error Correction Efficiency**: Assessed by the frequency with which participants no longer encounter syntax errors after using a hint or checking the exercise. Indicates how effectively participants apply provided assistance to correct errors.

Methodology

Standardization Each component was first standardized to allow comparability, ensuring that differences in scales did not disproportionately influence the composite index. Standardization was achieved using z-scores:

$$Z = \frac{(X - \mu)}{\sigma} \tag{3.1}$$

where X is the raw score, μ is the mean and σ is the standard deviation for each metric.

Weighting For this analysis, we assign equal weights to all components, reflecting their equal importance in assessing study success. The weights can be adjusted based on relative importance as determined by future analysis or expert input.

Computation of Composite Index The overall composite index was calculated using the following formula:

$$\begin{aligned}\text{Composite Index} = &\frac{1}{3} \times \text{Standardized Success Rate} \\ &+ \frac{1}{3} \times \text{Standardized Syntax Mistake Frequency} \\ &+ \frac{1}{3} \times \text{Standardized Error Correction Efficiency}\end{aligned} \tag{3.2}$$

Interpretation and Validation
The composite index provides a holistic view of study success:

- **Higher Scores**: Indicate superior overall performance, characterized by high success rates, reduced syntax mistakes, and effective error correction.
- **Validation**: We validated the composite index by correlating it with additional outcome measures to ensure it accurately reflects the construct of study success.

Sensitivity Analysis We conducted sensitivity analyses to evaluate how changes in the weights or individual components affect the robustness of the index. This ensures that the composite index remains a reliable tool in various scenarios.

By combining these key metrics into a single index, we can more effectively assess and compare the overall impact of interventions, identifying which strategies enhance learning outcomes more effectively.

3.12.6. RQ3: Moderating Variables

Research Question Three investigates the role of moderating variables in influencing self-efficacy and related metrics, with a specific focus on gender as a potential moderator.

Components of the ANCOVA Model

- **Independent Variable**: The intervention method, categorized into participants receiving LLM-based error messages versus those receiving conventional error messages.
- **Dependent Variable**: Students' post-intervention self-efficacy scores, measured through a standardized scale.
- **Covariate**: Pre-test self-efficacy scores, adjusted to control for initial differences across gender.
- **Moderating Variable**: Gender, assessed for its role in influencing the impact of the intervention on self-efficacy outcomes.

Methodology
Model Specification The ANCOVA model was specified to examine whether gender moderates the relationship between the intervention type and post-intervention self-efficacy outcomes, while controlling for pre-intervention self-efficacy levels. This model aims to identify the unique contribution of gender as a moderating variable.

Assumptions and Validation The assumptions of ANCOVA, including normality of residuals and homogeneity of regression slopes, were validated, particularly given the small sample sizes within gender categories. Any deviations from these assumptions were carefully evaluated.

Outcome Metrics The ANCOVA model provided estimates for:

- **Adjusted Group Means by Gender**: Showing the self-efficacy levels for each gender group after adjusting for baseline scores.
- **Effect Sizes**: Measured using Partial Eta-Squared to quantify the magnitude of gender's moderating effect.

- **Significance Testing**: Evaluating whether gender differences in the intervention's impact are statistically significant.

Interpretation and Implications

The results of the ANCOVA provide insight into the moderating effect of gender on educational interventions:

- **Differential Adjusted Means by Gender**: Suggests potential gender-based differences in how interventions impact self-efficacy, highlighting the importance of considering gender in educational strategies.
- **Significant Moderating Effects**: Indicate that gender may influence how interventions affect self-efficacy, suggesting a need for gender-sensitive educational practices.

Sensitivity Analysis To confirm the robustness of our findings, sensitivity analyzes were conducted, evaluating the results against variations in model assumptions and potential covariates. These analyses ensure that conclusions remain consistent under different analytic conditions, providing a comprehensive understanding of gender's moderating role in educational interventions.

4

Results

4.1. Filter Statistics

A total of 67 records were initially considered. After applying the exclusion criteria for participants under 16 years of age (20 participants) and having too much prior knowledge (5 participants), we filtered down to 32 usable records, maintaining 47.8% of the original sample.

4.2. Demographics

The final sample consisted of 32 participants, divided into a control group (n = 19) and a test group (n = 13). The gender distribution was 16 females, 15 males, and 1 identified as none of the above.

4.3. Syntax Error Analysis

Out of 679 requests for hints or exercise checks, 199 syntax errors were identified. The errors were predominantly Mismatched Close Tags (55.78%), followed by Unclosed Tags (14.07%) and Close Tags for Void Elements (13.57%). The distribution of syntax error types is summarized in Table 4.1.

Table 4.1: Syntax Error Type Distribution

Error Type	Frequency (%)
Mismatched Close Tag	55.78
Unclosed Tag	14.07
Close Tag for Void Element	13.57

4.4. Markov Chain Analysis

When comparing the Markov chain analysis (Figure 4.1) between the test and control groups, it is evident that both groups navigated the application in the same way. Upon requesting a hint, it is observed that the control group more frequently exhibited syntax errors rather than other types of errors. In contrast, the test group encountered other errors more frequently than syntax errors. However, these observations are limited in their conclusiveness because the transitions are expressed in relative terms and do not account for absolute values or uncertainty.

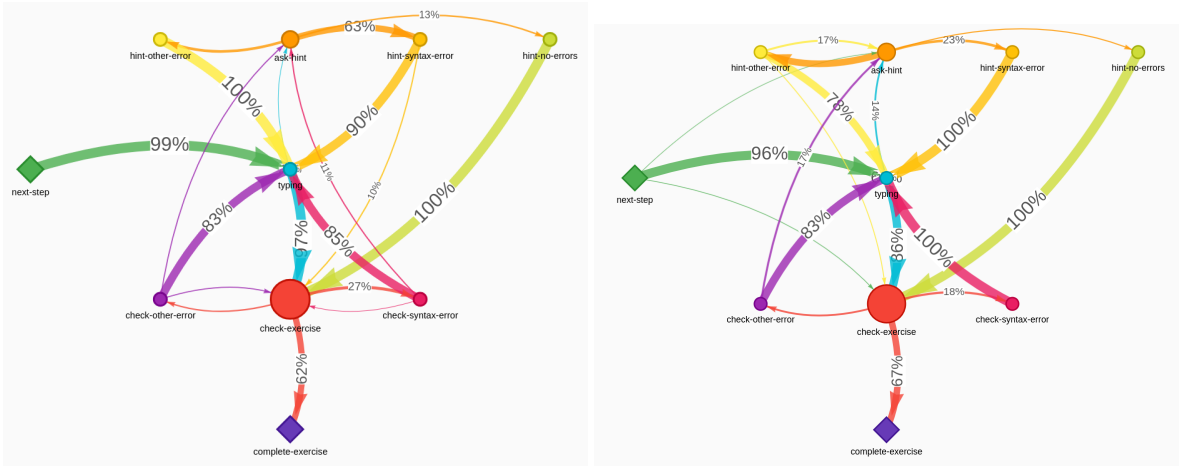


Figure 4.1: Markov chain analysis for the control group (left) and the test group (right).

4.5. RQ1: Self-Efficacy

An ANCOVA did not reveal significant differences in post-test self-efficacy scores between the groups when controlling for pre-test scores, $F(1, 28) = 1.80, p = .191$. The test group had a slightly higher adjusted mean (3.485) compared to the control group (3.786), but this difference was not statistically significant. The effect size was medium ($\eta^2 = 0.0603$).

A visual representation of the regression lines for pre-test and post-test self-efficacy scores across control and test groups can be seen in Figure 4.2. Detailed ANCOVA results are provided in Table 4.2.

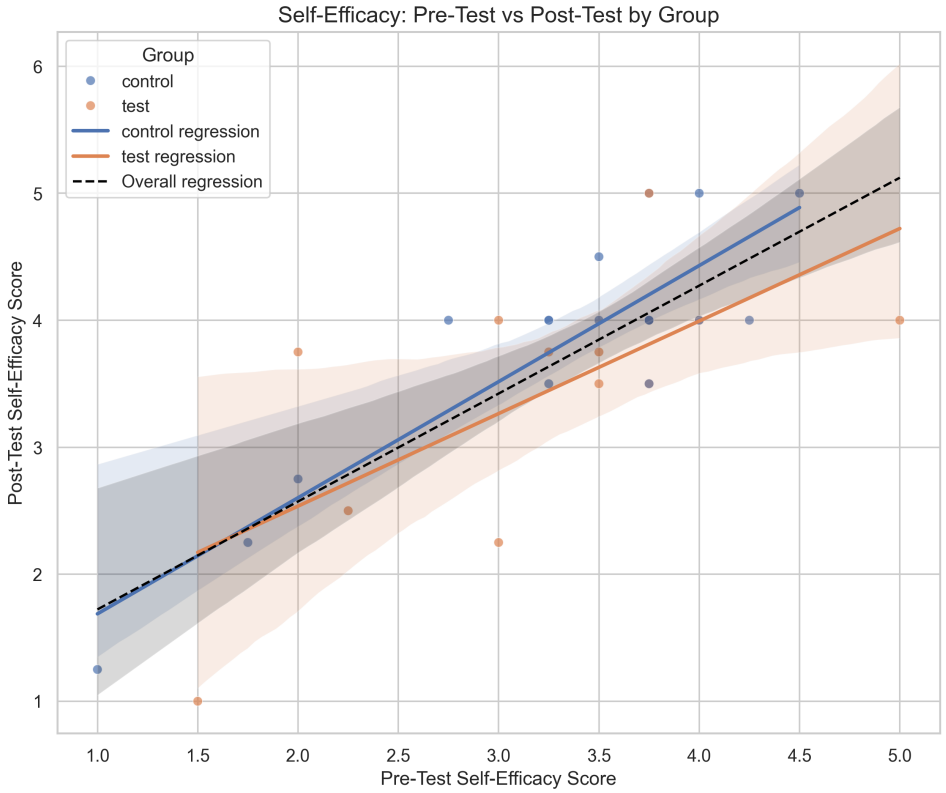


Figure 4.2: Results of an ANCOVA test showing regression lines for pre-test and post-test self-efficacy scores across control and test groups, with overall trend analysis.

	Sum of Squares	df	F-value	p-value
Group	0.659	1	1.80	0.191
Pre-test Self-Efficacy	17.023	1	46.39	< 0.001
Residual	10.275	28		

Table 4.2: ANCOVA results for post-test self-efficacy scores controlling for pre-test scores.

4.5.1. Adjusted Means and Effect Sizes

The adjusted post-test means controlled for pre-test scores were 3.786 for the control group and 3.485 for the test group as shown in Table 4.2. The partial eta-squared for the group effect was $\eta^2 = 0.0603$, indicating a medium effect size.

4.5.2. Normality Assumption Tests

Due to the small sample size per group, the normality assumptions for ANCOVA were tested using the Shapiro-Wilk test. The results indicated that the pre-test and post-test scores were not normally distributed for the control group, while they were normally distributed for the test group. The findings underscore the need to interpret the results with caution given the potential impact on the robustness of the ANCOVA. The distributions can be seen in Figure 4.3.

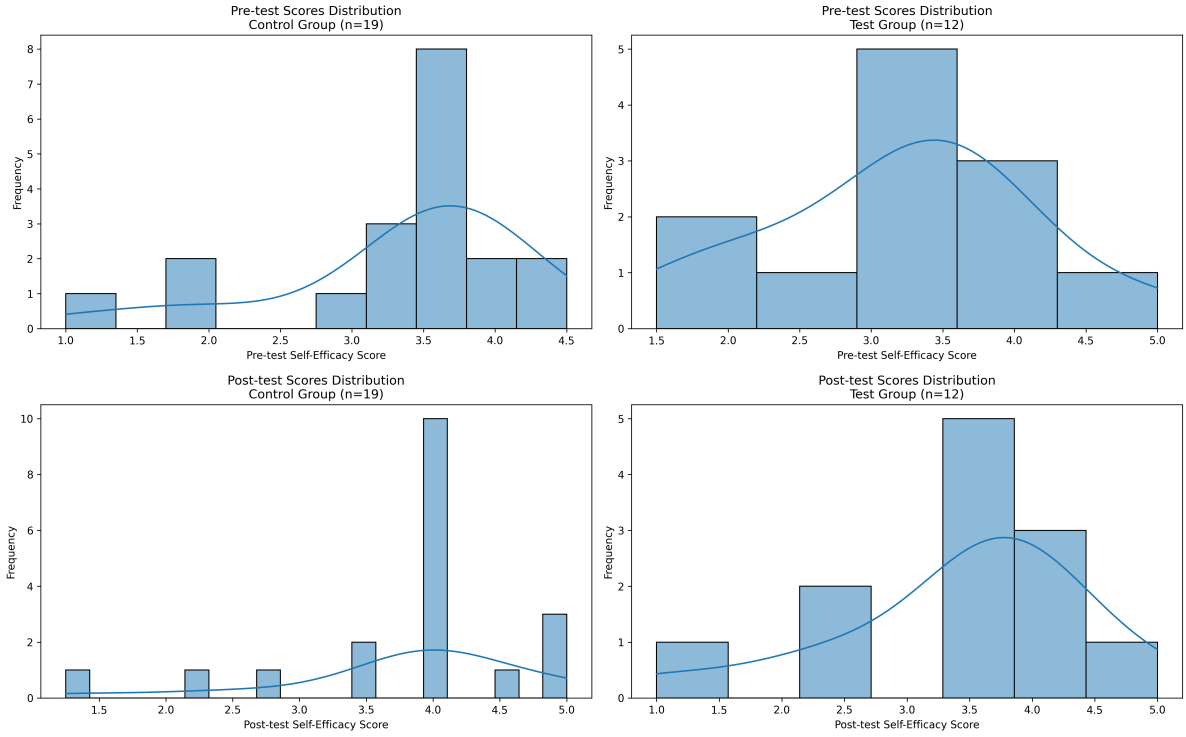


Figure 4.3: Histograms showing the distribution of pre-test and post-test self-efficacy scores by group.

4.5.3. Hypothesis Testing

Hypotheses:

- **H0:** There is no significant relationship between the use of a Large Language Model (LLM) for rewriting HTML syntax error messages and students' self-efficacy compared to conventional error messages.
- **H1:** There is a significant positive relationship between the use of a Large Language Model (LLM) to rewrite HTML syntax error messages and students' self-efficacy compared to conventional error messages.

Conclusion: Based on the ANCOVA results, there were no significant differences in self-efficacy between the test and control groups when controlling for pre-test scores. Therefore, we fail to reject the null hypothesis (**H0**) and conclude that the use of a Large Language Model (LLM) does not have a significant relationship with students' self-efficacy compared to conventional error messages.

4.6. RQ2: Study Success

4.6.1. Composite Study Success Index

The composite study success index was constructed to evaluate the overall performance of the participants by integrating three primary metrics: success rate, syntax mistake frequency, and error correction efficiency. Each metric was standardized using z-scores, and an equal weighting scheme was initially applied. Sensitivity analysis was performed to examine the impact of varying the weights between different components.

Comparison Between Test and Control Groups

The analysis involved comparing the composite index between the test and control groups. Here are the summary statistics and results from the t-test for the composite index under the equal weighting scheme, as presented in Table 4.3.

Group	Mean	Std	Count	Min	Max
Control	0.0314	0.9527	19	-2.0532	1.0934
Test	-0.0597	0.6698	10	-1.0793	0.6673

Table 4.3: Composite Index Statistics by Group

Figure 4.4 shows a boxplot visualization of the composite study success index by group.

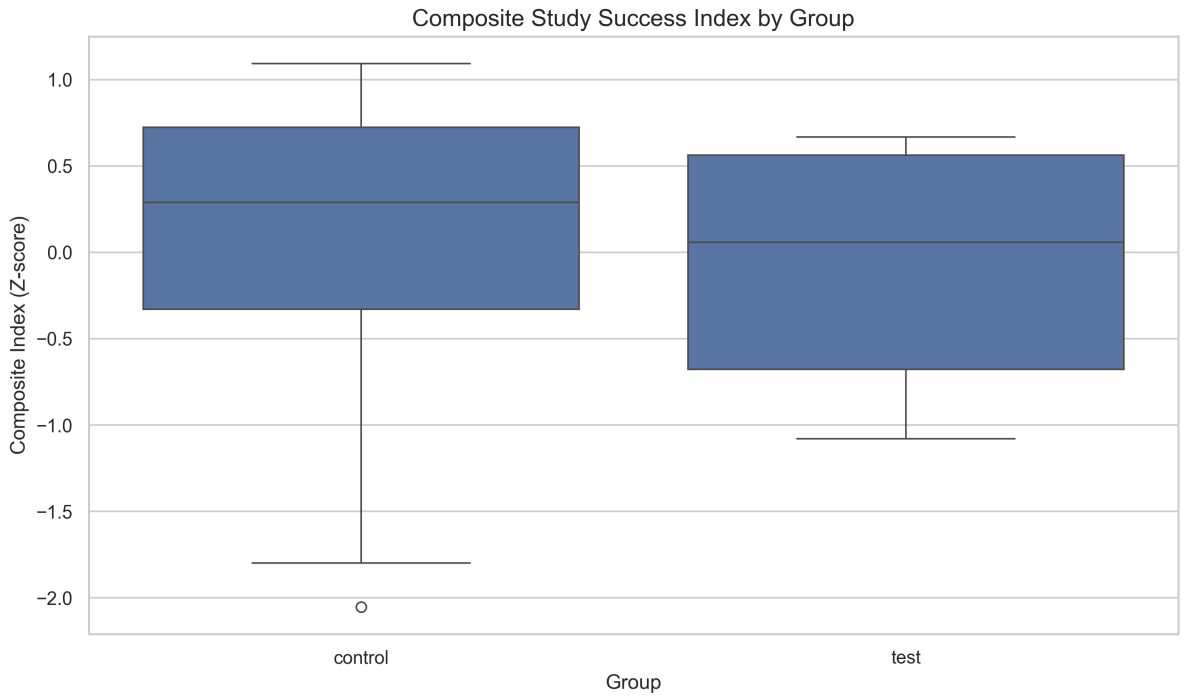


Figure 4.4: Boxplot of Composite Study Success Index by Group

The results of t-test between the test and control groups did not show significant differences in the composite index (t-statistic: -0.2993, p-value: 0.7673). The effect size, measured by Cohen's d, was -0.1049, interpreted as negligible.

Sensitivity Analysis

The sensitivity analysis tested various weighting schemes for the components of the composite index. None of the alternative weighting schemes resulted in a significant difference between the test and control groups, as summarized in Table 4.4.

Weighting Scheme	Mean Difference	p-value	Significant
Equal weights	-0.0911	0.7673	No
Emphasis on success rate	-0.1251	0.7012	No
Emphasis on syntax mistakes	-0.0396	0.8969	No
Emphasis on error correction	-0.1086	0.7238	No
Success and syntax only	-0.0562	0.8741	No
Success and correction only	-0.1941	0.5537	No
Syntax and correction only	-0.0231	0.9391	No

Table 4.4: Sensitivity Analysis Summary

Figure 4.5 provides a visual representation of the sensitivity analysis, showing mean differences between test and control groups.

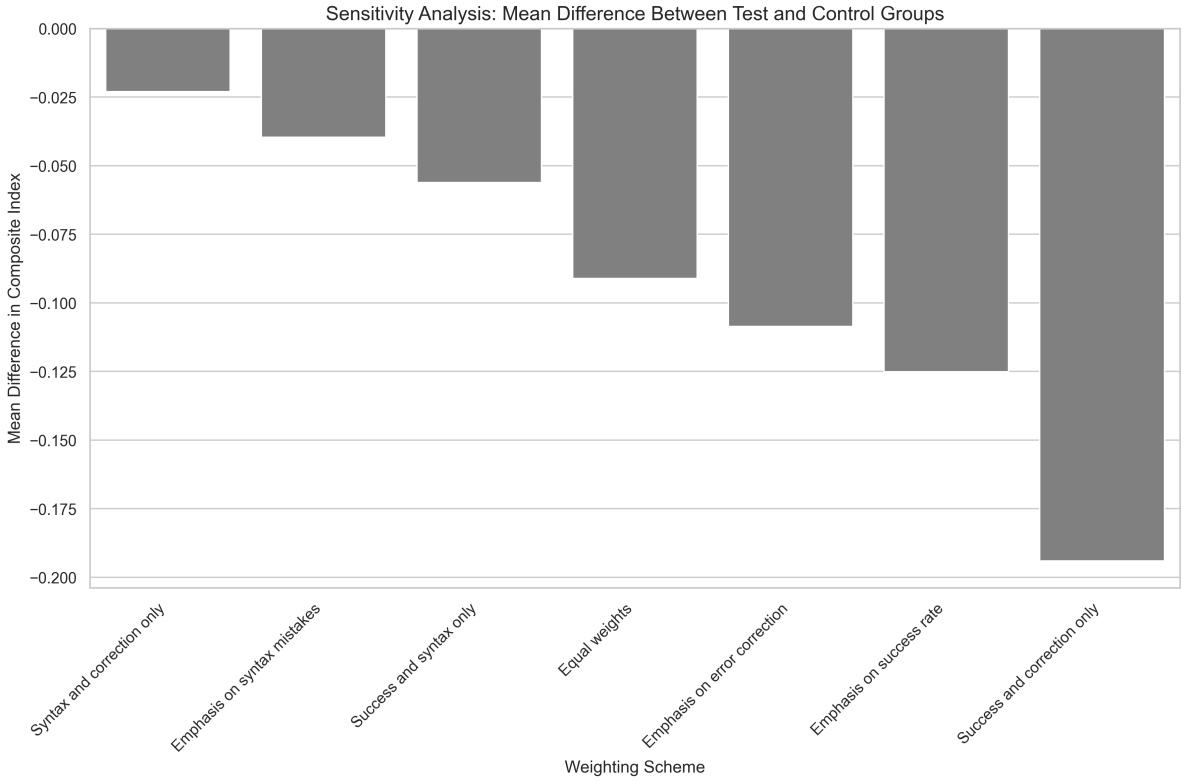


Figure 4.5: Sensitivity Analysis: Mean Difference Between Test and Control Groups

4.6.2. Scatter Matrix of Components and Composite Index

A scatter matrix (Figure 4.6) illustrates the relationships between the components of the composite study success index and the index itself. This visualization helps to understand the distribution and correlations among the metrics between different groups.

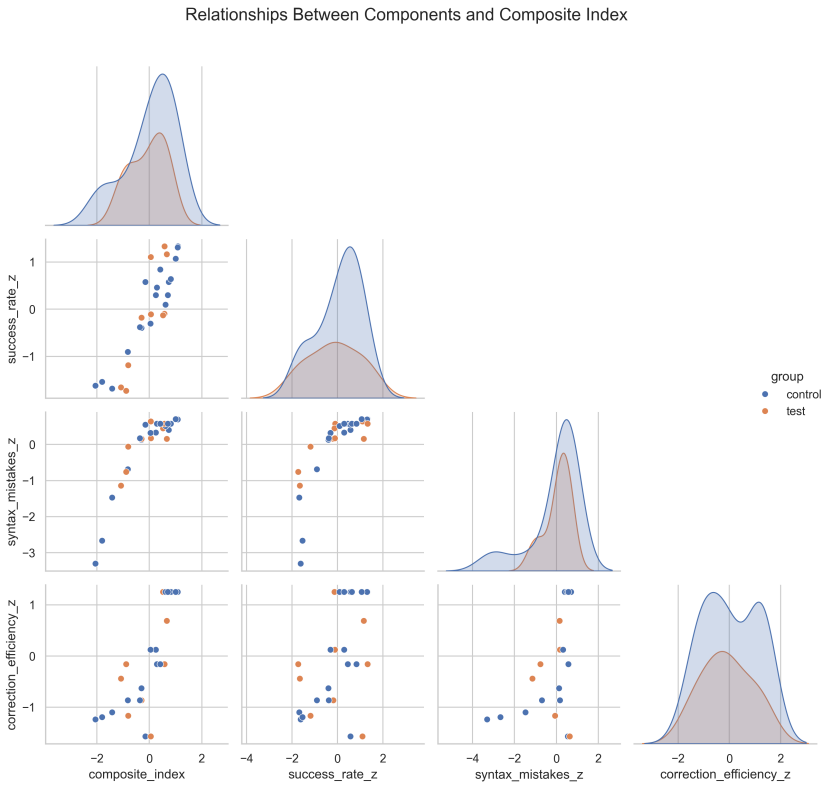


Figure 4.6: Scatter Matrix of Components and Composite Index

4.6.3. Hypothesis Testing

Hypotheses:

- **H0:** The integration of a Large Language Model (LLM) in rewriting HTML syntax error messages does not have a significant relationship with study success compared to conventional error messages.
- **H1:** The integration of a Large Language Model (LLM) in rewriting HTML syntax error messages has a significant positive relationship with study success compared to conventional error messages.

Conclusion: Based on the composite index analysis, there were no significant differences in study success between the test and control groups. Therefore, we fail to reject the null hypothesis (**H0**) and conclude that the integration of a Large Language Model (LLM) does not have a significant relationship with study success compared to conventional error messages.

4.7. RQ3: Moderating Variables

The analysis examined whether gender moderates the relationship between intervention type (LLM-based error messages vs. conventional error messages) and post-intervention self-efficacy outcomes, controlling for pre-intervention self-efficacy scores.

4.7.1. ANCOVA Results

An ANCOVA was conducted to explore the moderating role of gender on the impact of interventions on self-efficacy. The results indicated no significant main effect of group ($F(1, 26) = 1.60, p = .218$), gender ($F(1, 26) = 1.41, p = .245$), or interaction between group and gender ($F(1, 26) = 0.04, p = .839$) on post-test self-efficacy scores when pre-test self-efficacy scores were controlled. The partial eta-squared for the interaction effect was very small ($\eta^2 = 0.0016$).

Figure 4.7 shows an interaction plot depicting adjusted post-test self-efficacy means by group and

gender. Detailed ANCOVA results for this analysis can be found in Table 4.5.

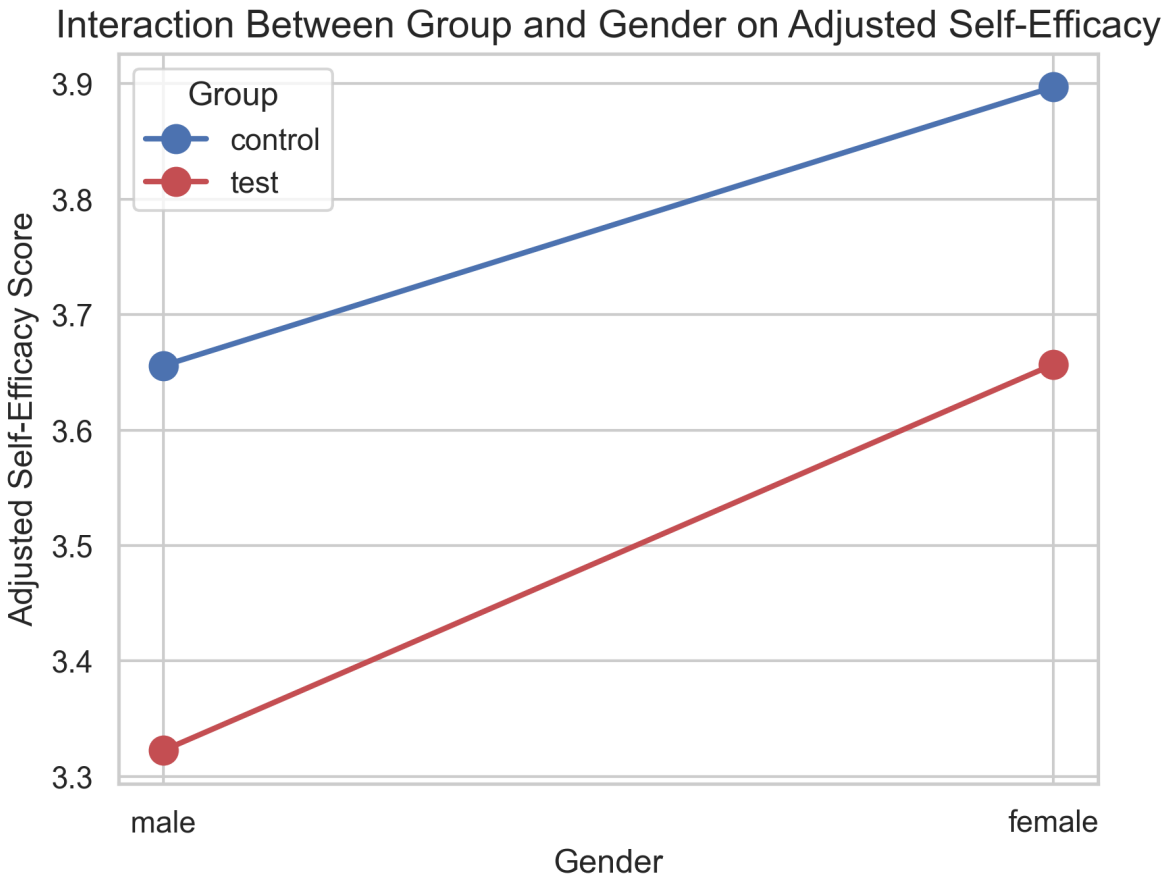


Figure 4.7: Interaction plot showing adjusted post-test self-efficacy means by group and gender.

	Sum of Squares	df	F-value	p-value
Group	0.598	1	1.60	0.218
Gender	0.529	1	1.41	0.245
Interaction	0.016	1	0.04	0.839
Pre-test Self-Efficacy	17.103	1	45.70	< 0.001
Residual	9.730	26		

Table 4.5: ANCOVA results examining the effect of group, gender, and their interaction on post-test self-efficacy scores, controlling for pre-test scores.

4.7.2. Adjusted Means and Effect Sizes

The adjusted post-test means controlled for pre-test scores were 3.786 for the control group and 3.485 for the test group, as shown in Table 4.5. The partial eta-squared for the group effect was $\eta^2 = 0.0603$, indicating a medium effect size. For gender, the effect size was $\eta^2 = 0.0515$, classified as a small effect.

4.7.3. Normality Assumption Tests

Normality assumptions for ANCOVA were evaluated using the Shapiro-Wilk test. For the control group, pre-test and post-test scores were not normally distributed, while they were normally distributed for the test group. These results suggest that the findings should be interpreted with caution due to potential impacts on the robustness of the ANCOVA. The distributions are visualized in Figure 4.8.

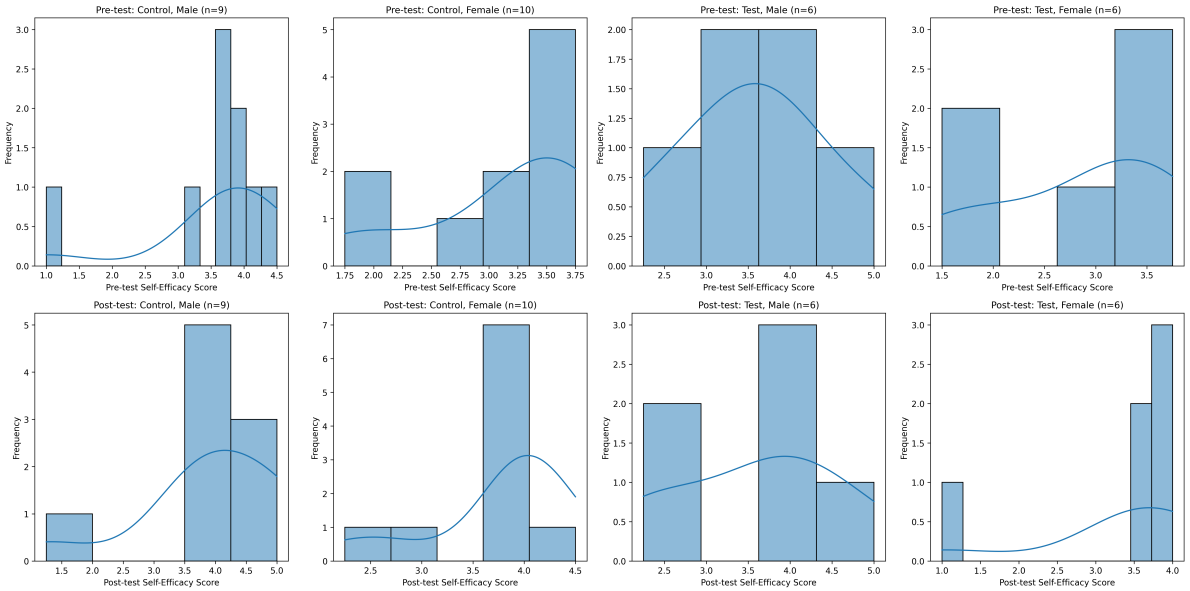


Figure 4.8: Histograms showing distribution of pre-test and post-test self-efficacy scores by group and gender.

4.7.4. Hypothesis Testing

Hypotheses:

- **H0:** Gender does not moderate the relationship between the intervention type and self-efficacy outcomes.
- **H1:** Gender significantly moderates the relationship between the intervention type and self-efficacy outcomes.

Conclusion: Based on the ANCOVA results, there was no significant interaction effect between the group and gender on self-efficacy. Therefore, we fail to reject the null hypothesis (**H0**) and conclude that gender does not significantly moderate the relationship between intervention type and self-efficacy outcomes.

5

Discussion

In examining the study findings, we explore the factors that influence the results, focusing primarily on self-efficacy, study success, and a critical analysis of the limitations encountered. The observed lack of statistical significance necessitates a reconsideration of potential contributing factors. It is plausible that the sample size of the participating students was insufficient to reveal significant differences; including more participants could have amplified the effect of the intervention, thus reducing the noise of the results. Furthermore, the intervention may not have been robust enough to elicit significant changes, as it was only triggered by syntax errors, potentially rendering the intervention minimal. Additionally, the comparison with Slowparse, which already provides more user-friendly error messages, may have diminished the observable advantage of the LLM rewritten error messages. It is also possible that the platform being used effectively provided immediate feedback, making error messages less critical in positively influencing the coding experience. Given that Bitsized already offered a live preview of the code and instant automated feedback, the coding experience may have already been significantly positive compared to traditional teacher-led methods, leading to insignificant results. In addition, no tests were performed on the prompt itself. Therefore, refinement of the prompt might produce more favorable outcomes. A well-engineered prompt could potentially offer a greater advantage over Slowparse's error messages. Despite these considerations, our findings align with those documented by [Santos and Becker, 2024], underscoring the ongoing discourse on the complexities of evaluating educational interventions.

The primary focus of our study was to measure self-efficacy and study success, although it did not evaluate knowledge retention or skill enhancement, specifically regarding HTML proficiency. The emphasis on self-efficacy aligns with the understanding that an individual's belief in their capabilities can profoundly impact learning outcomes. However, it is essential to consider how these beliefs might translate into practical skills. Reflections of Klopfer, as noted in [Shein, 2024], strongly support the central themes of this research. Klopfer asserts that the process of struggle and effort is crucial in learning: "Working hard and struggling is actually an important way of learning. When you're given an answer, you're not struggling and you're not learning. And when you get more of a complex problem, it's tedious to go back to the beginning of a large language model and troubleshoot it and integrate it." This viewpoint highlights that while clearer error messages might offer better guidance, the learning process is unlikely to see substantial enhancement without incorporating struggle and active problem-solving engagement.

5.1. Limitations

The limitations of this study include the fact that it was conducted exclusively with students from one school in the Netherlands, which may affect the generalizability of the results. The sample size was insufficient to ensure normal distributions for both the test and control groups with respect to the pre- and post-self-efficacy scores, which could compromise the robustness of the ANCOVA. The use of a large language model introduces variability that may hinder the exact reproduction of results when faced with the same syntax errors. The testing environment was not completely silent, as there were

instances where the students assisted each other despite our efforts to minimize such interactions.

Future research should aim to address these limitations by including a more diverse participant pool, increasing the sample size, ensuring a controlled testing environment, and standardizing the use of large language models to improve the reliability and reproducibility of the findings.

6

Conclusion

In conclusion, this study aimed to explore the effects of Large Language Model (LLM)-assisted feedback on students' self-efficacy and study success in learning HTML compared to conventional error messages provided by Slowparse. The analysis was carried out with a filtered sample of 32 high school students, evenly distributed between genders and divided into control and test groups.

The findings reveal no significant differences in self-efficacy between the test and control groups, as assessed through ANCOVA, controlling for pre-test scores. Although the test group showed slightly lower self-efficacy mean scores compared to the control group, the effect size was not substantial. This indicates that LLM-assisted feedback did not enhance self-efficacy more than the conventional feedback from Slowparse.

Regarding study success, although the control group achieved a slightly higher success rate in completing HTML exercises compared to the test group, the differences were found to be statistically insignificant according to the analysis of the t-test. Furthermore, no significant differences were observed in syntax error correction rates between the two groups, indicating that the feedback form, whether LLM-assisted or conventional, did not affect the efficiency of error correction.

In exploring the moderating variables, particularly gender, the analysis did not show notable effects or interactions with the study outcomes. With balanced gender distributions between groups, the findings did not reveal significant variations in response behaviors related to self-efficacy between male and female participants.

Overall, while LLM-assisted feedback was hypothesized to positively impact self-efficacy and study success, the experimental results did not support these expectations. These findings suggest that the integration of an LLM for providing feedback, as configured in this study, did not provide advantages over syntax error messages from Slowparse in improving students' motivation or learning outcomes in this context.

7

Future Work

Based on the findings of this study, future research should explore the application of LLM support in a variety of programming languages. This expansion could provide valuable information on whether the efficacy of error message interventions varies with different coding environments, thereby enhancing our understanding of the generalizability of this support.

A critical area for further investigation involves optimizing the specific prompts used to trigger LLM interventions. This involves exploring different instructional approaches, such as Socratic and Didactic methods, which could potentially influence how learners engage with the material. Furthermore, investigating the impact of emotional support provided by the LLM, whether empathetic or non-empathetic, could offer significant insights into the user experience. As detailed in [Breazeal et al., 2024], these elements could substantially affect how users perceive and interact with error messages, potentially making them less frustrating and more educational.

By tailoring prompts to different teaching styles and emotional support frameworks, we can advance our understanding of the role of LLMs in education. This research could lead to the development of more adaptive and responsive educational tools that not only address syntax errors effectively but also enhance student engagement and learning satisfaction.

Based on the data collected in this research, there are opportunities for qualitative analysis. For example, experts could analyze the code with syntax mistakes and evaluate the helpfulness of the LLM-generated messages. This could determine whether these messages are really beneficial, similar to the method used in [Leinonen et al., 2023]. Furthermore, the data could be further analyzed to identify specific cases of students who experienced the highest gains or losses in self-efficacy.

References

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychol. Rev.*, *84*(2), 191–215.

Bandura, A. (1982). Self-efficacy mechanism in human agency. *American Psychologist*, *37*(2), 122–147. https://doi.org/10.1037/0003-066X.37.2.122

Bandura, A. (1997). *Self-efficacy: The exercise of control*. Freeman.

Beyer, S. (2014). Why are women underrepresented in computer science? gender differences in stereotypes, self-efficacy, values, and interests and predictors of future cs course-taking and grades. *Computer Science Education*, *24*(2-3), 153–192. https://doi.org/10.1080/08993408.2014.963363

Breazeal, C., Rai, A., Ramesh, B., Chen, L., Long, Y., Aria, A., Loi, H., Torralba, A., Bernstein, J., Reich, J., Klopfer, E., Abelson, H., Westerman, G., & Bosch, C. (2024). Opportunities, Issues, and Challenges for Generative AI in Fostering Equitable Pathways in Computing Education [https://mit-genai.pubpub.org/pub/sy1uboa4]. *An MIT Exploration of Generative AI*.

Campbell, J. C., Hindle, A., & Amaral, J. N. (2014). Syntax errors just aren't natural: Improving error reporting with language models. *Proceedings of the 11th Working Conference on Mining Software Repositories*, 252–261. https://doi.org/10.1145/2597073.2597102

Eltegani, N., & Butgereit, L. (2015). Attributes of students engagement in fundamental programming learning, 101–106. https://doi.org/10.1109/ICCNEEE.2015.7381438

Fowler Jr, F. J., & Cosenza, C. (2009). Design and evaluation of survey questions. *The SAGE handbook of applied social research methods*, *2*, 375–412.

Hasan, B. (2003). The influence of specific computer experiences on computer self-efficacy beliefs. *Computers in Human Behavior*, *19*(4), 443–450. https://doi.org/https://doi.org/10.1016/S0747-5632(02)00079-1

Khan. (n.d.). Khan/live-editor: A browser-based live coding environment. https://github.com/Khan/live-editor

Kleppang, A. L., Steigen, A. M., & Finbråten, H. S. (2023). Explaining variance in self-efficacy among adolescents: The association between mastery experiences, social support, and self-efficacy. *BMC Public Health*, *23*(1), 1665.

Leinonen, J., Hellas, A., Sarsa, S., Reeves, B., Denny, P., Prather, J., & Becker, B. A. (2023). Using large language models to enhance programming error messages. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 563–569. https://doi.org/10.1145/3545945.3569770

Mauriello, N., Pagnucci, G. S., & Winner, T. (1999). Reading between the code: The teaching of html and the displacement of writing instruction. *Computers and Composition*, *16*(3), 409–419. https://doi.org/https://doi.org/10.1016/S8755-4615(99)00020-1

Mozilla. (n.d.). Mozilla/slowparse: A slow js-based html parser with good error feedback and debugging metadata. https://github.com/mozilla/slowparse

Park, T. H., & Wiedenbeck, S. (2011). Learning web development: Challenges at an earlier stage of computing education. *Proceedings of the Seventh International Workshop on Computing Education Research*, 125–132. https://doi.org/10.1145/2016911.2016937

Phillips, R. S., & Brooks, B. P. (2017). The hour of code: Impact on attitudes towards and self-efficacy with computer science. *Code: Seattle, WA, USA*.

Santos, E. A., & Becker, B. A. (2024). Not the silver bullet: Llm-enhanced programming error messages are ineffective in practice. *Proceedings of the 2024 Conference on United Kingdom & Ireland Computing Education Research*. https://doi.org/10.1145/3689535.3689554

Saris, W. E., Revilla, M., Krosnick, J. A., & Shaeffer, E. M. (2010). Comparing questions with agree/disagree response options to questions with construct-specific response options. *Survey Research Methods. 2010; 4 (1): 61-79*.

Schunk, D. H. (1987). Self-efficacy and cognitive achievement.

Schwarzer, R. (1995). Generalized self-efficacy scale. *Measures in health psychology: A user's portfolio. Causal and control beliefs/Nfer-Nelson*.

Self-efficacy in discipline scale. (n.d.). Retrieved January 4, 2025, from https://www.imperial.ac.uk/research-and-innovation/education-research/evaluation/what-can-i-evaluate/self-efficacy/tools-for-assessing-self-efficacy/self-efficacy-in-discipline-scale/

Shein, E. (2024). The impact of ai on computer science education. *Commun. ACM*, *67*(9), 13–15. https://doi.org/10.1145/3673428

Smith, S., Kass, S., Rotunda, R., & Schneider, S. (2006). If at first you don't succeed: Effects of failure on general and task-specific self-efficacy and performance. *North American Journal of Psychology*, *8*, 171–182.

Teeuw, B., Schwarzer, R., & Jerusalem, M. (1994). Dutch general self-efficacy scale. *Retrieved November*, *22*, 2010.

van Maanen, A. (2021, September). *Self-efficacy meten bij jonge leerlingen op de basisschool: Ontwikkeling van een valide en betrouwbaar meetinstrument om self-efficacy van dag tot dag te meten bij jonge leerlingen* [Master's thesis]. Universiteit van Amsterdam [Faculteit der Maatschappij- en Gedragswetenschappen, Opleiding Master Onderwijswetenschappen]. https://scripties.uba.uva.nl/search?id=c5113790

Wang, S., Mitchell, J., & Piech, C. (2024). A large scale rct on effective error messages in cs1. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 1395–1401. https://doi.org/10.1145/3626252.3630764

Wright, J. D. (1975). Does acquiescence bias the" index of political efficacy?". *The Public Opinion Quarterly*, *39*(2), 219–226.

Zimmerman, B. J. (1995, April). Self-efficacy and educational development. In A. Bandura (Ed.), *Self-Efficacy in changing societies* (pp. 202–231). Cambridge University Press.

A

Self-Efficacy Survey HTML Module

Beantwoord de volgende vragen terwijl je over jezelf nadenkt als een leerling die deze HTML-cursus volgt.

1. Hoe zeker ben je dat je al het werk kunt voltooien dat is toegewezen in de module over HTML?
    - Helemaal niet zeker
    - Een beetje zeker
    - Enigszins zeker
    - Vrij zeker
    - Volledig zeker
2. Hoe zeker ben je dat je complexe ideeën in de HTML-module kunt begrijpen?
    - Helemaal niet zeker
    - Een beetje zeker
    - Enigszins zeker
    - Vrij zeker
    - Volledig zeker
3. Hoe zeker ben je dat je al het materiaal in de HTML-module kunt leren?
    - Helemaal niet zeker
    - Een beetje zeker
    - Enigszins zeker
    - Vrij zeker
    - Volledig zeker
4. Hoe zeker ben je dat je het moeilijkste werk in de HTML-module kunt uitvoeren?
    - Helemaal niet zeker
    - Een beetje zeker
    - Enigszins zeker
    - Vrij zeker
    - Volledig zeker
5. Hoe zeker ben je dat je volgend jaar nog weet wat je hebt geleerd in de huidige HTML-module?
    - Helemaal niet zeker
    - Een beetje zeker

- Enigszins zeker
- Vrij zeker
- Volledig zeker

B

Consent Form & Research Information

B.1. Consent Form & Introduction to Research

You are invited to participate in a study titled 'Research on Self-Efficacy and Study Outcomes in Learning HTML'. This research is conducted by Kris van Melis from TU Delft, in collaboration with Bitsized Publisher BV, where Kris van Melis is also a co-owner.

The purpose of this study is to investigate the impact of LLM-assisted feedback on self-efficacy and study outcomes when learning HTML. LLM stands for 'Large Language Model', a type of artificial intelligence specialized in processing and generating human language. The study will take approximately 45 minutes. The collected data will be used for scientific publications and educational development.

You are asked to answer a number of questions before and after the lesson on HTML, and to complete exercises in HTML utilizing LLM-assisted feedback. The following data will be collected:

- **Survey-response data**: Collection of responses from pre- and post-study surveys to assess changes in students' self-efficacy and to analyze demographic differences.
- **Platform analytics data**: Consisting of anonymized collected interactions with the learning platforms, such as number of attempts per exercise, button usage data, inactive and active time data, problem solving speed, and success rate data.

As with any online activity, there is the risk of a data breach. We do our utmost to keep your data confidential. The data is collected anonymously and stored. Personal data such as IP addresses or other identifiable information are not collected. Personal identity data, such as your name, are not retained. Once the study has been completed, the anonymized data will be made public via 4TU.ResearchData for further scientific purposes.

Your participation in this study is completely voluntary, and you can stop answering questions at any time without consequences. However, once the study is completed, you no longer have the option to withdraw your participation as the data is collected anonymously.

There is no financial compensation or grade associated with your participation in this study. Additionally, your teacher will not have access to your personal data.

For questions or complaints, please contact the responsible researcher:
Kris van Melis

By participating in the study and proceeding to the survey, you indicate that you have read the consent form and agree with it.

B.2. Consent Formulier & Introductie tot Onderzoek

Je wordt uitgenodigd om deel te nemen aan een onderzoek genaamd 'Onderzoek naar Zelfeffectiviteit en Studieresultaten bij het Leren van HTML'. Dit onderzoek wordt uitgevoerd door Kris van Melis van de TU Delft, in samenwerking met Bitsized Publisher BV, waarbij Kris van Melis ook mede-eigenaar is.

Het doel van dit onderzoek is om de impact van LLM-geassisteerde feedback op zelfeffectiviteit en studieresultaten te bestuderen bij het leren van HTML. LLM staat voor 'Large Language Model', een type kunstmatige intelligentie gespecialiseerd in het verwerken en genereren van menselijke taal. Het onderzoek zal ongeveer 45 minuten in beslag nemen. De verzamelde data zullen worden gebruikt voor wetenschappelijke publicaties en onderwijsontwikkeling.

Je wordt gevraagd om een aantal vragen te beantwoorden vóór en na de les over HTML, en om oefeningen in HTML te voltooien waarbij gebruik wordt gemaakt van LLM-geassisteerde feedback. De volgende data zal worden verzameld:

- **Survey-respons data**: Verzamelen van responses uit de pre- en post-studie enquêtes om veranderingen in zelfeffectiviteit van studenten te beoordelen en demografische verschillen te analyseren.
- **Platform analytics data**: Bestaande uit geanonimiseerd verzamelde interacties met de leerplatforms, zoals aantal pogingen per oefening, knopgebruik data, inactieve en actieve tijdsdata, probleemoplossingssnelheid, en succescijfer data.

Zoals bij elke online activiteit is er het risico van een datalek. Wij doen ons uiterste best om je gegevens vertrouwelijk te houden. De gegevens worden anoniem verzameld en opgeslagen. Persoonlijke data zoals IP-adressen of andere identificeerbare informatie worden niet verzameld. Persoonlijke identiteitsgegevens, zoals je naam, worden niet bewaard. Nadat de studie is afgerond, worden de geanonimiseerde data openbaar gemaakt via 4TU.ResearchData voor verdere wetenschappelijke doeleinden.

Je deelname aan dit onderzoek is volledig vrijwillig, en je kunt op elk moment stoppen met het beantwoorden van vragen zonder gevolgen. Echter, zodra de studie is afgerond, heb je de optie niet meer om je deelname in te trekken doordat de data volledig anoniem wordt verzameld.

Er is geen financiële vergoeding of cijfer verbonden aan je deelname aan dit onderzoek. Daarnaast zal je docent geen toegang krijgen tot jouw persoonlijke data.

Voor vragen of klachten kun je contact opnemen met de verantwoordelijke onderzoeker:
Kris van Melis
k.p.vanmelis@student.tudelft.nl

Door deel te nemen aan het onderzoek en naar de survey door te klikken, geef je aan dat je het consent formulier hebt gelezen en hiermee instemt.

B.3. Information Letter for Parents

Dear Parents/Guardians,

Your child is invited to participate in a study titled "*Research on Self-Efficacy and Study Outcomes in Learning HTML*," conducted by Kris van Melis from TU Delft. Within this study, your child will learn the basis of how to build websites using HTML.

Purpose of the study:
The study aims to investigate how feedback provided by Large Language Models (LLMs), a type of artificial intelligence, affects students' confidence and learning outcomes in HTML. The study will take about 45 minutes. The data collected will help in scientific publications and educational development.

What your child will do:

- Answer questions before and after an HTML lesson.
- Complete HTML exercises with the help of LLM-assisted feedback.

Data collection:

- Survey responses to measure changes in self-confidence and analyze demographic differences.
- Anonymous platform data on usage patterns like attempts per exercise and problem-solving speed.

Data privacy:
All data is collected anonymously. Personal information like names or IP addresses is not collected. The anonymized data will be made public for further scientific purposes.

Voluntary participation:
Your child's participation is entirely voluntary. They can stop at any time without consequences, but withdrawal is not possible after completion as data is anonymous.

Contact:
For questions or concerns, please contact the researcher:

Kris van Melis at `k.p.vanmelis@student.tudelft.nl`

B.4. Informatiebrief voor ouders

Beste Ouders/Verzorgers,

Uw kind is uitgenodigd om deel te nemen aan een onderzoek genaamd "*Onderzoek naar Zelfeffectiviteit en Studieresultaten bij het Leren van HTML*," uitgevoerd door Kris van Melis van de TU Delft. Tijdens dit onderzoek zal je kind de basis leren om websites te bouwen met HTML.

Doel van het onderzoek:
Het onderzoek beoogt te bestuderen hoe feedback van grote taalmodellen (LLMs), een type kunstmatige intelligentie, het zelfvertrouwen en leerrendement van studenten in HTML beïnvloedt. Het onderzoek duurt ongeveer 45 minuten. De verzamelde data helpen bij wetenschappelijke publicaties en onderwijsontwikkeling.

Wat uw kind zal doen:

- Vragen beantwoorden voor en na een HTML-les.
- HTML-oefeningen maken met behulp van LLM-geassisteerde feedback.

Gegevensverzameling:

- Enquêteresponses om veranderingen in zelfvertrouwen te meten en demografische verschillen te analyseren.
- Anonieme platformdata over gebruikspatronen zoals pogingen per oefening en probleemoplossingssnelheid.

Gegevensprivacy:
Alle gegevens worden anoniem verzameld. Persoonlijke informatie zoals namen of IP-adressen worden niet verzameld. De geanonimiseerde data worden voor verdere wetenschappelijke doeleinden openbaar gemaakt.

Vrijwillige Deelname:
Deelname van uw kind is volledig vrijwillig. Ze kunnen op elk moment stoppen zonder gevolgen, maar intrekking is na voltooiing niet mogelijk, omdat de gegevens anoniem zijn.

Contact:
Voor vragen of opmerkingen, neem contact op met de onderzoeker:

Kris van Melis op `k.p.vanmelis@student.tudelft.nl`