

ECS Web Lab: A Web-Based Solution for Equitable and Engaging Web Development Education

Mr. Samuel B Mazzone, Marquette University

Dr. Dennis Brylow, Marquette University

ECS Web Lab: A Web-Based Solution for Equitable and Engaging Web Development Education

Abstract

UCLA developed the Exploring Computer Science (ECS) curriculum to increase participation of women and people of color in computer science, focusing on content comprehension, inquiry-based learning, and educational equity. Initially launched in Los Angeles, ECS expanded nationwide with NSF support. However, teachers in Milwaukee encountered challenges in adapting the ECS curriculum, particularly Unit 3 on web development, due to virtual settings, rising truancy, and Chromebook limitations. Many teachers opted for web-based alternatives with automated coding activities, despite these being less effective for underrepresented students. These platforms lacked project submission and feedback features, increasing teacher workload.

In response, we developed the ECS Web Lab (ECS-WL), a Chromebook-compatible web tool designed to facilitate HTML and CSS learning while preserving ECS's core principles: CS comprehension, inquiry-based learning, and educational equity.

ECS-WL integrates features such as an in-browser code editor using CodeMirror, secure HTML sanitization with Dompurify, and real-time data management via Firebase. This tool enables students to write, test, and submit code directly through their browsers, eliminating the need for software installations and addressing logistical hurdles faced by educators.

To enhance ECS-WL's pedagogical effectiveness, we revised Unit 3 of the ECS curriculum. The modifications introduced assignments that encouraged students to create web pages based on their personal interests and individual experiences, promoting equity and engagement. For instance, students designed websites about their favorite restaurants or fictional businesses, allowing them to express their identities and cultures authentically. These adjustments clarified guidance, reduced ambiguity, and supported inquiry-based, student-driven projects.

During the 2023-2024 academic year, ECS-WL was piloted in five classrooms within the MPS district, involving 242 students who created and submitted 755 unique web pages using 41 distinct HTML tags. Assessments of student submissions on the first two assignments revealed a wide array of topics reflecting personal interests and unique perspectives. Students created websites about local, chain, and fictional restaurants representing various ethnic cuisines, and a diverse selection of books and movies. The Linguistic Inquiry and Word Count (LIWC) tool showed that students added more personal content, authenticity, and references to their motivations and lifestyles compared to the starter code, indicating deeper engagement and

incorporation of their experiences.

However, the pilot identified areas for improvement. Student engagement declined noticeably by Day 9, before the introduction of CSS on Day 10, suggesting the curriculum's length and structure may be overwhelming. Teachers noted ECS-WL's inability to customize the curriculum, manage multiple classes, or facilitate group work. Presenting the entire 25-day curriculum at once was daunting, and the absence of features like a class overview menu and language translation options limited the tool's accessibility and usability.

In response, we have implemented several enhancements to ECS-WL. Teachers can now edit daily instructions for classroom customization. A system to hide complete and future days has made the curriculum more flexible, allowing length adjustments. Features to distinguish between different ECS classes taught by the same teacher and to group students for collaborative learning have improved classroom management. An overview menu offers a comprehensive view of student progress. Additionally, a personal workspace enables students to apply daily concepts to their projects, fostering creativity. We will collect teacher feedback through surveys on engagement, usability, curriculum adaptability, and student outcomes. Student progress will be tracked through metrics like assignment completion rates and evaluations of submitted work. This feedback will guide ongoing refinements to ECS-WL and the curriculum.

Introduction

To increase the participation of women and people of color in computer science (CS), UCLA developed the ECS curriculum¹ for use in the Los Angeles Unified School District. ECS serves both as a high school introductory CS curriculum and a professional development (PD) program for educators. ECS is built on three core principles: CS content comprehension, inquiry-based learning, and educational equity². This approach ensures that CS education is accessible and engaging for underrepresented students, fostering a supportive and equitable learning environment¹. With the help of multiple National Science Foundation grants in the 2010s, ECS spread to many other regions of the country.

The COVID-19 pandemic significantly impacted education globally beginning in 2020, propelling a widespread shift towards remote work and education. Communities using ECS implemented remote teacher workshops to address the challenges exposed by the pandemic³, but in Milwaukee many teachers struggled with the transition, and ECS implementation rates began to drop sharply in MPS. Our investigation uncovered multiple reasons that teachers were turning from the rigors of ECS to competing alternatives. Teachers struggled to convert the rich group work of ECS into virtual and hybrid modalities. Strapped for time, they found web-based curricular alternatives with slick, pre-recorded instructional videos, and highly automated coding activities to be alluring, despite the research showing lower engagement and success among precisely those students most supported by ECS. Meanwhile, skyrocketing truancy rates in MPS (58% of students chronically absent in 2021-22, attending less than 90% of school days⁴; MPS students attended only 70% of the time) undermined sequential progression in coursework, especially in curricula like ECS unit 3. Finally, while ubiquitous use of Chromebooks in MPS simplifies many aspects of device administration, the difficulty in installing many kinds of standard software presented additional challenges for teachers during web development (WD). As a result, many

teachers turned to websites offering valuable WD materials and coding environments requiring no installation. However, these platforms lacked features for direct student project submission and teacher project review, in effect further sapping scarce teacher time and effort.

Our challenge was developing a Chromebook-compatible web tool for HTML and CSS learning that would directly address many of the logistical and pedagogical pain points raised up by our ECS classroom practitioners, while still preserving the core aspects of the ECS curriculum's culturally-responsive and inquiry-led pedagogy. Our response was to build the ECS Web Lab (ECS-WL) and its corresponding modifications to Unit 3 of the ECS version 9 curriculum and PD. The new PD was deployed with teachers in October 2023, and ECS-WL was piloted by those teachers in MPS classrooms in the following winter and spring. The deployment of ECS-WL is a major prong in our efforts to re-engage and revitalize our ECS teacher community after the challenges of the pandemic. In this paper we present the design and major features of ECS-WL and evaluate the results of its pilot year in MPS.

The research questions guiding this experiment were:

1. Can students effectively utilize the ECS-WL tool to create and submit websites to their teachers?
2. Do the assignments replacing existing ECS curriculum activities demonstrate cultural responsiveness and provide opportunities for students to express their personal interests?

Prior and Related Work

Exploring Computer Science

In 2016, Chicago Public Schools (CPS) mandated CS as a graduation requirement⁵. ECS course enrollments at CPS rose significantly, from 3,165 in 2013 to 7,901 in 2017, and 15,220 in 2021⁶.

The growing numbers of ECS teachers, students and schools have experienced numerous benefits. A previous study found that 43% of ECS students enrolled in additional high school CS courses, compared to only 26% of students who started with alternative classes⁷. Furthermore, a post-ECS survey revealed a significant surge in student interest in learning more about CS, with 43% expressing being “very interested,” up from 17% before the program began⁸. Students who have completed CS courses, particularly ECS, also tend to earn higher grades in subsequent science courses than students who have not completed CS courses⁹. A disparate impact analysis of the statewide deployment of ECS in nearby Wisconsin showed that increased access to CS was significantly greater in schools in the lowest socioeconomic quartile¹⁰.

ECS PD offers significant benefits for educators, with 91% finding PD sessions useful². Even during COVID-19 remote implementations, 95% of educators reported that ECS PD strengthened their understanding of inquiry-based teaching methods³.

The ECS curriculum is constantly being improved and extended, both by its original authors and other stakeholders. The tenth edition of ECS is being released for the 2024-25 school year. The MUzECS project¹¹ introduced a low-cost module using Arduino Leonardo and a custom block-based programming language, replacing the expensive module in the final unit of ECS

version 6. A unit on sewn electronic textile projects¹² focused on iterative design in computational thinking. Students engage in hands-on learning with programmable circuits using sensors and actuators, helping them embrace mistakes and build understanding and resilience.

However, the curriculum's impact on WD proficiency has been less pronounced for both students and teachers. A study done in CPS found that students demonstrate the lowest mastery in WD compared to other areas¹³. Another study tracking coaching interactions amongst ECS teachers revealed that Unit 3 required the highest rate of coaching interactions¹⁴, underlining that this unit frequently presents pain points for classroom practitioners.

Importance of Web Development

Debates took place in Milwaukee regarding the observed teacher shift away from teaching HTML and CSS in ECS Unit 3. The decision to advocate for continuing the teaching of WD was driven by several factors. WD offers students a platform to create diverse projects while gaining foundational CS knowledge. Past instructors highlighted the transferable skills learned through WD, and the fulfillment derived from creative autonomy and project ownership¹⁵. Additionally, research has demonstrated that using HTML and CSS in introductory programming courses increased student engagement and retention by allowing them to create personalized web pages¹⁶. This approach helped students grasp fundamental programming concepts in a meaningful way, especially those with little prior experience.

Insights in Web Development Education

ECS-WL builds upon insights and effective strategies that have emerged from prior tools and research. In one instance of an introductory WD course, students encountered challenges with document types, embedding links and images, and understanding page layout, including positioning and alignment¹⁷. This study also suggested that a WD course focusing on HTML and CSS first and saving JavaScript for later may better serve students – particularly those without prior experience – to avoid confusion and negative attitudes toward programming.

In WD, openHTML¹⁸ offered valuable insights as an educational tool. Its live previews helped students quickly identify and resolve errors. However, browsers rendered buggy code correctly, reinforcing misunderstandings. Combining live previews with validation and linting tools can create a more robust educational framework, promoting accurate coding skills¹⁹.

Another study, also using openHTML, found that approximately 20% of student errors are due to the complex rules dictating valid nesting of HTML elements, while 35% of errors are related to simpler tag syntax mistakes. This underscores the importance of mastering both the fundamental and intricate aspects of HTML syntax²⁰. Despite its significant contributions to WD education, openHTML was discontinued. The website remained accessible until April 16, 2016, before it was taken down.

Web Development: Other Existing Tools

In the field of WD education, numerous modern tools and platforms have been established to facilitate the learning process for students. These platforms provide comprehensive resources and interactive environments that help learners grasp HTML and other web technologies. The following are some notable tools that have significantly contributed to WD education:

Feature	W3 Schools	Learn-HTML.org	CodeCademy	Web.Dev	ECS-WL
Teacher Submission	✗	✗	✗	✗	✓
Includes Subscription Barrier	✓	✗	✓	✗	✗
Interest-Driven Project-Based Learning	✗	✗	✗	✗	✓
Advertisements	✓	✓	✗	✗	✗
Coding Assessment Feedback	—	—	✓	✗	✓

Figure 1: Other Web Development Tools vs. ECS-WL

W3Schools²¹ is a popular WD learning platform with tutorials and references for HTML, CSS, JavaScript, and other web technologies. It features interactive code examples and a “Try it Yourself” feature for editing and executing code in the browser. The platform also includes fill-in-the-blank exercises that engage users in active learning. However, these exercises only accept one correct answer and lack personalized feedback. Extra exercises, quizzes, certifications, and ad-free access require a subscription (see Figure 1).

Learn-HTML.org²² is a free tool for learning HTML and CSS through lessons with explanations, examples, and interactive exercises. Solutions are revealed upon clicking a designated button. However, the tool uses hard-coded solutions, limiting user personalization and providing minimal feedback.

Codecademy²³ is a well-known online learning platform that offers interactive coding courses on various programming languages and web technologies. Codecademy’s engaging and interactive courses feature hands-on exercises, quizzes, and real-world projects, all within a user-friendly, gamified platform. While this tool provides only one solution to the exercises, it offers immediate feedback on code submissions, helping students correct errors and understand their next steps. However, it requires a subscription to access features beyond the lessons, such as quizzes and projects.

Google’s Web.Dev²⁴ is a comprehensive free resource covering a broad range of WD topics, along with resources and tutorials on HTML and CSS. Web.dev includes articles, code examples, and occasional interactive code labs. However, the labs only provide code demonstrations, not guided projects, and the quiz feature offers no assessment or feedback on coding ability.

Methodology

Students in MPS use Chromebooks, complicating the process of installing third-party apps. As a response, basic HTML/CSS compilation was implemented into ECS-WL, allowing students to write and test their code directly in their browsers. To create an effective and accessible learning environment, we relied on robust, user-friendly platforms: Firebase for database management, CodeMirror for code editing, and Dompurify for enhanced security (see Figure 2 for platform overview).

Firestore: Cloud-Based Database Solution

Firestore²⁵ is a platform developed by Google for creating mobile and web applications. It provides a variety of tools and infrastructure designed to simplify the development process. Firestore's database allows developers to store and synchronize data between users in real-time, making it an ideal solution for applications that require instantaneous data updates.

Key features of Firestore include authentication and Cloud Firestore. Firestore offers a robust authentication system, which supports various authentication methods, including email and password, phone authentication, and third-party providers like Google and Facebook. For MPS students, this means they can use their existing MPS-managed Gmail accounts for authentication, streamlining the login process. Cloud Firestore is a flexible and scalable real-time NoSQL cloud database. It offers live synchronization and offline support, providing a seamless user experience even when connectivity is intermittent. Relying upon existing cloud tools for the development of ECS-WL provides a more robust and secure tool than more homegrown solutions.

CodeMirror: Text Editor for Code

CodeMirror²⁶ is an open-source JavaScript library that provides a versatile text editor specifically designed for editing code. It supports a wide range of programming languages and offers various features that make it a valuable tool for WD.

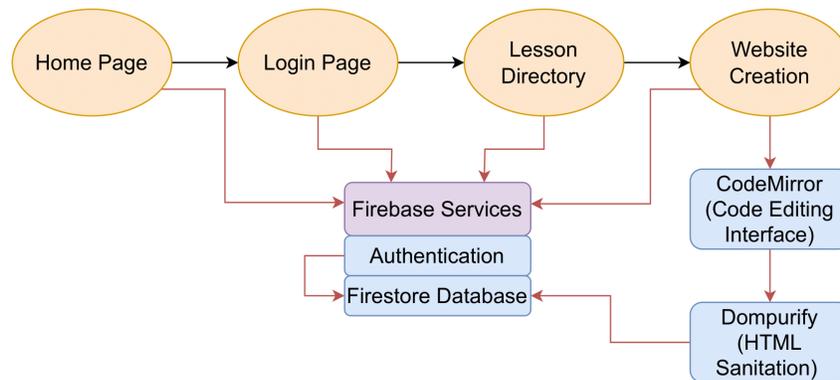


Figure 2: The Interaction of 3rd Party Tools within ECS-WL

Some of the key features of CodeMirror include syntax highlighting and line numbering. Syntax highlighting helps users easily distinguish between different elements of their code, while line numbering helps to more quickly identify and reference specific lines when debugging. These

features are crucial for both novices and experienced developers alike, as it makes the code more readable and reduces the likelihood of errors.

During the development of ECS-WL, CodeMirror was implemented to enhance the HTML and CSS coding experience for students. By integrating CodeMirror, students were provided with a more sophisticated and user-friendly code editor.

Dompurify: Secure HTML Sanitization

Dompurify²⁷ is a powerful library for sanitizing HTML and CSS to prevent XSS (cross-site scripting) attacks. XSS is a vulnerability that allows attackers to inject and execute malicious scripts in users' browsers. Dompurify cleans user-generated content before insertion into the HTML document, enhancing web application security.

Dompurify provides thorough sanitization by removing malicious code from HTML and CSS, such as scripts, iframes, and embedded objects. This ensures code runs and outputs safely. Additionally, Dompurify allows developers to customize the sanitization process by specifying allowed and disallowed tags and attributes.

By using Dompurify, the ECS-WL website can ensure that all user-generated content is thoroughly sanitized before being displayed, thereby protecting against potential XSS attacks. Dompurify also allows only HTML and CSS code to run, blocking all JavaScript and any other language from being run. This library ensures the output is safe from malicious scripts and is required only on the web creation page. The cleaned code is sent to the Firebase database.

Improving Student Code Management

ECS-WL incorporates a "Get Starter Code" button to directly address one of the ECS teachers' pain points. This was designed to provide specific foundational code tailored to each day's activity, without the need for the teacher to develop an independent delivery mechanism. Template starter code was prepared for each day of the modified curriculum, offering more extensive guidance at the beginning to assist students as they start their WD journey. Two more buttons were introduced to enhance functionality. The "Save and Turn In Assignment" button enables users to save their work and submit the day's assignment. The "Get Previously Saved Code" button allows users to retrieve their previously saved web page.

Teacher feedback during beta testing of ECS-WL highlighted a mismatch between the unmodified curriculum and ECS-WL. The existing ECS curriculum provided solid content, but the associated activities fell short. It instructed students to apply HTML and CSS concepts to their websites without direct assignments, resulting in confusion due to the broad scope and lack of guidance on content.

Curriculum Adjustments for ECS-WL

The existing ECS curriculum was updated to better align with the ECS-WL. Previously, the lesson plan covered HTML and CSS concepts. The revised curriculum retains these concepts but enhances learning by requiring students to create web pages based on their interests. This approach aimed to encourage student research and provides specific instructions, making the tasks less ambiguous.

The curriculum development process began with identifying equitable WD topics that would allow all students to embrace their own interests, identities, or cultures. Culturally responsive and sustaining curriculum design is a key pillar of equitable CS education²⁸. Topics include creating websites on various interests, such as favorite animals or board games. They also cover practical scenarios, like selecting items for their school garden or writing articles as if contributing to their school newspaper. Later in the curriculum, students design websites for fictional yet realistic businesses, such as an event organizing service or a photography business.

Legend: HTML Concepts CSS Concepts

Lesson Number	Concept Covered	Website Premise
1	doctype, html, head, title, body, h1-h6, p	Favorite Restaurant
2	img, a	Favorite Book
3	b, strong, i, em	Green Space
4	Learning Check 1	Favorite Animal
5	ul, ol, li, hr, br	Favorite Holiday Dish
6	table, thead, tbody, tr, th, td	Phone Comparison
7	blockquote, q, abbr	School Newspaper
8	Learning Check 2	Recycling
9	style, font-size, color	Dream Vacation
10	link, border-style, background-color	Music Playlist
11	border-color, border-width, height/width units, color units	Morning Routine
12	Learning Check 3	Historical Event
13	padding, margin	Future Career
14	div, width, height,	Favorite TV Show
15	section, article, classes	Ideal Weather
16	Learning Check 4	Favorite Celebrity/Figure
17	line-height, letter-spacing, font-weight, font-style	Favorite Drink
18	text-transform, text-align, text-decoration	Favorite Board Game
19	list-style-position, list-style-type	Event Organizer
20	Learning Check 5	Nonprofit/Charity
21	a:link, a:visited, a:hover, a:active	Fitness Studio
22	display:block, display:inline, display:inline-block, display:flex, display:none	Travel Agency
23	position:static, position:relative, position:fixed, position:absolute	Photography Studio
24	Learning Check 6	Themed Wedding
25	Final Assignment	What Culture Means To You

Table 1: Unit 3 Overview Using ECS-WL

Next, these topics of interest were integrated into the existing ECS curriculum. Originally, the ECS curriculum included lessons that spanned multiple days. This structure was revised so that each day a new webpage would be created. Consequently, additional HTML and CSS concepts were incorporated into the curriculum. For each day of the curriculum, “Key Concepts,” “Notes On Topics,” and “Instructions For Assignment” buttons and text were created. The concepts and notes were organized through bullet points, and the instructions were organized numerically to offer a clear guide on what concepts and content students need to include on their webpages. The instructions thoroughly elaborate on certain topics to help prevent common errors that researchers have identified novice web developers often encounter^{17,19}. The starter code was also updated to

match each day of this new curriculum. Each day was aligned with CSTA K-12 Standards.

To illustrate the result, assignment one will be examined. For this assignment, students choose either their favorite restaurant or create their own. They research the items on the restaurant’s menu and build a web page featuring these items. Through this assignment, they learn about HTML metadata, headers, and paragraphs.

An additional objective was to promote the intelligent use of AI tools while preventing students from becoming overly reliant on them for their assignments. The curriculum, designed around students’ personal interests, achieved this by ensuring that although students can use AI tools like ChatGPT²⁹ for assistance, they cannot simply copy and paste assignment instructions without engaging with the material. Blindly copying and pasting produces generic responses that do not reflect the unique interests of each student, making it easy for teachers to identify a lack of thoughtful engagement. If students do choose to use AI for assistance, they must engage in thoughtful prompt engineering to tailor the AI’s output to their personal criteria. This approach guarantees that with or without the use of AI, every piece of work reflects genuine personal engagement and critical thinking.

To assess students’ understanding of the concepts without using starter code, we introduced learning check days. These days do not cover any new material. Instead, they evaluate students’ knowledge of previously taught concepts. On these days, the buttons “Key Concepts,” “Notes On Topic,” and “Get Starter Code” are removed. Students are encouraged to review previous lessons to complete the project for these days.

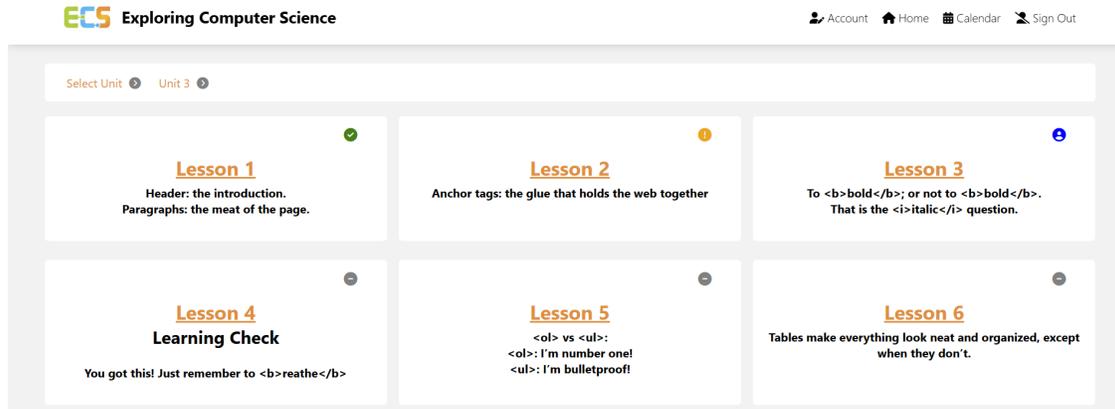


Figure 3: The Layout of the ECS-WL Website: Directory Page

Enhanced Web Development Learning

A key addition to the ECS-WL was an error detection feature. If an HTML or CSS error is found, a red icon appears on the affected line, clearly indicating the location of the issue to students. Our approach aligns with findings from previous studies, which emphasize the importance of integrating error detection tools to prevent the reinforcement of faulty coding understandings¹⁹.

Another significant improvement involved the implementation of realistic CSS linking. Logic was introduced to give students the flexibility to name the CSS file as they prefer, but also ensure that

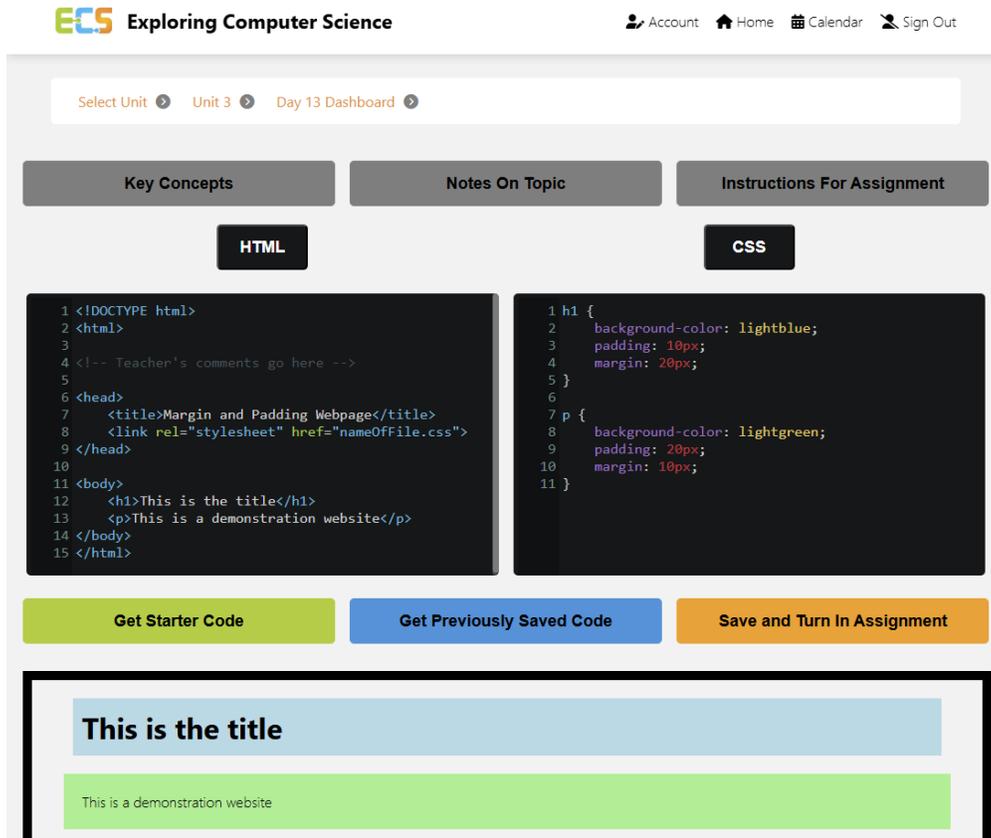


Figure 4: The Layout of the ECS-WL Website: Website Creation

separate CSS files function correctly only if the appropriate linkage code is provided. As previous research has indicated, this can be a challenging area for students¹⁷. Therefore, we provided detailed and thorough instructions for this aspect of WD.

Support for multi-page websites was added to ECS-WL. This enhancement was achieved by providing starter code for linking to other pages and adjusting the directory for effective navigation. This update expanded the scope of student projects and enriched their learning with real-world WD practices.

As seen in Figure 3, the website was also improved by implementing a notification system. Teachers can now add notifications to student websites during the grading process, allowing for real-time progress monitoring. A visual tracker uses icons to represent daily progress: “gray” for not started, “blue” for pending teacher review, “orange” for student review needed, and “green” for no further edits required. This system simplifies grading for teachers and provides a clear visual representation of student progress.

Results

During the 2023-2024 semester, five teachers integrated the ECS-WL tool into their classrooms. A total of 242 students used this tool, resulting in the creation and submission of 755 unique

webpages. Students submitted webpages using 41 distinct valid HTML tags.

To assess whether students were incorporating elements that reflected their individual creativity when developing their submitted webpages, two methods were employed. The first method involved manually reviewing the topics chosen on days one and two and analyzing those results. The second method used an automated process with LIWC (Linguistic Inquiry and Word Count)³⁰, a text analysis tool.

Manual Assessment

Days one and two were selected for manual feedback extraction due to their high volume of submissions. Day one has students create a webpage about either an existing restaurant or one of their own design. Students research the items that would be on the restaurant's menu. Restaurant title was extracted manually for each student submission, and categorized as local, part of a chain, or fictional. If the chosen restaurant was real, its ethnicity was reported. For fictional restaurants, the ethnicity was extracted only if it was explicitly mentioned in the name of the restaurant.

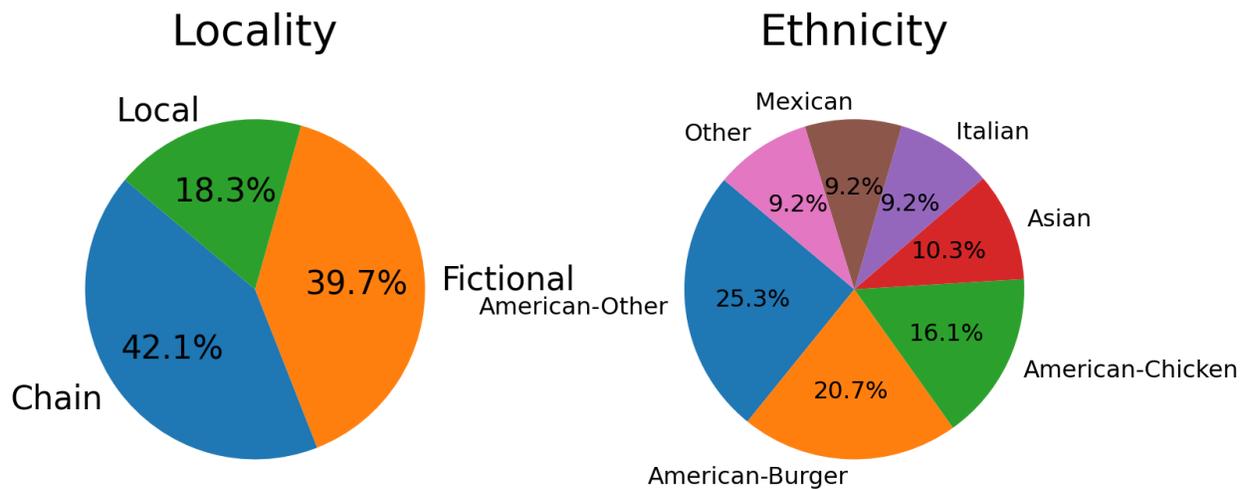


Figure 5: Diverse Student Choice for Day 1 Activity

Out of 134 entries, 126 entries contained an extractable restaurant name. Out of these valid submissions, 101 unique restaurants were identified, accounting for 80.2%. Among the valid submissions, 42.1% were categorized as part of a chain, 39.7% as fictional, and 18.3% as local, as depicted in Figure 5.

Out of 126 entries with a valid title, 87 (69.0%) had an extracted ethnicity. This was done either by finding the ethnicity of the real restaurant or extracting the ethnicity in the name of the fictional restaurant. Figure 5 shows that restaurant ethnicity also demonstrated a wide distribution. Out of 87 web pages, American-Other (restaurants with diverse menus not primarily focused on burgers or chicken,) made up 25.3% of the entries. American-Burger accounted for 20.7%. American-Chicken represented 16.1%. Asian restaurants comprised 10.3%, while both Mexican and Italian restaurants each accounted for 9.2%. Ethnicities in the "other" category include Puerto Rican, Greek, Vietnamese, and African restaurants, as well as restaurants that feature multiple ethnicities. The "other" category made up 9.2% of all entries.

Day two's project required students to choose their favorite book or movie and create a webpage based on their selection. For each submission, the title of the book or movie was extracted.

Day two had 101 total entries, of which 96 had an extractable title. Among the 96 entries, the distribution between books and media was distinctly skewed. Books accounted for 80.2% of the total selections, with 77 out of 96 entries. Within the category of books, 52 unique book titles were identified, representing 67.5% of the total book entries, as illustrated in Figure 6. A unique book title refers to a distinct book series or standalone book, where even different books within the same series are considered as one unique book title.

On the other hand, media entries constituted 19.8% of the total, with 19 out of 96 entries, each entry unique. In total, 74.0% of all day two submissions topics were uniquely chosen.

To the extent to which it is possible to quantitatively gauge creativity and freedom of expression in this type of classroom activity, ECS-WL lessons one and two supported a pleasingly wide array of student-chosen preferences and cultural referents.

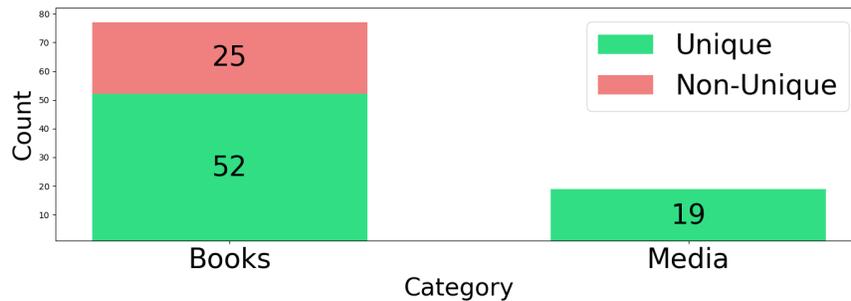


Figure 6: Diverse Student Choice for Day 2 Activity

Automated Assessment

LIWC is a text analysis tool that quantifies linguistic and psychological features in written text, providing insights into emotional, cognitive, and social processes through word usage patterns. In this study, an LIWC analysis was conducted on the text extracted from within the HTML tags of both the starter code provided to students and the websites submitted by students.

For the student submissions, values for each term in the LIWC dictionary were calculated. The day of each submission was tracked, and the number of submissions per day was recorded. Days with fewer than five submissions or designated as learning checks were excluded from further analysis. This exclusion applied to both the student submissions and the starter code.

To ensure reliability, a weighted average for each LIWC term was calculated, assigning greater importance to days with higher numbers of submissions. Similarly, the starter code results were averaged, with weights corresponding to the number of student submissions on the same days. This approach ensures that the comparisons between student work and starter code reflect the varying levels of student participation.

There were 12 eligible days that met this requirement totaling 601 out of the 755 student webpage submissions. As indicated in Table 2, the student webpage's weighted average word count (WC) according to the LIWC output was 72.6, significantly higher than the starter code WC of

Table 2: Comparing Starter Code and Student Submissions

	WC	Authentic	Drives	Lifestyle
Starter	25.6	20.5	2.0	4.6
Student	72.6	27.8	3.1	5.3

25.6.

In the LIWC dictionary, the term “Authentic,” which was originally based on studies of honesty and deception^{31,32} reflects the degree of self-monitoring, with high scores found in spontaneous, uninhibited conversations and low scores in prepared or socially cautious texts. The student average authentic score was 27.8, compared to the starter code’s 20.5. “Drives,” also in the LIWC dictionary, encompasses various motivations and needs expressed in the text. This score is higher in student submissions (3.1) than in the starter code (2.0). “Lifestyle” in the LIWC dictionary references daily activities and personal concerns. The student average lifestyle score was 5.3, compared to the starter code’s 4.6.

In summary, automated quantitative sentiment assessment of hundreds of student-developed pages showed significant investment by students in sharing an authentic voice.

However, student engagement with ECS-WL fell short of expectations. By Day 9, 602 out of 755 submissions (79.74%) had already been made before CSS was introduced on Day 10.

Discussion

The ECS-WL pilot was successful in its objectives, while also identifying areas for improvement. It effectively streamlined student WD, offering an accessible platform for learning HTML and CSS. Features such as Firebase for real-time database management, CodeMirror for the coding interface, and Dompurify for security enhanced functionality and reliability. The streamlined submission and review processes made it easier for teachers to monitor submissions and provide feedback efficiently.

The evaluation of the ECS-WL tool revealed that while teachers found it significantly more useful than previous solutions, they also identified several critical areas for improvement. One prominent issue was that teachers could not customize the new curriculum for their specific classrooms directly on the ECS-WL website. As a result, teachers had to provide modified instructions either on paper or through online methods. Another issue was the overwhelming presentation of the entire 25-day curriculum at once, which could be daunting for students. Additionally, the tool did not adequately differentiate between separate ECS classes taught by the same teacher, creating challenges in classroom management. The absence of functionality for grouping students further hindered the ability to track and manage group work, a crucial aspect of collaborative learning. Teachers also observed a decline in student engagement as the course progressed, raising concerns about the curriculum’s length. Moreover, the lack of an overview menu to view all students simultaneously and the absence of a toggle feature to translate the website into Spanish were highlighted as significant shortcomings. Lastly, teachers suggested adding a workspace where students can apply daily concepts to their personal projects.

The tool successfully incorporated CS concepts, inquiry, and equity into the platform. Students demonstrated their understanding of CS concepts through the extensive use of HTML tags. Inquiry-based learning was evident from the LIWC analysis, which showed students adding more personal content, authenticity, and references to their motivations and lifestyle activities. This indicates deeper engagement and understanding, as students are not only completing assignments but also integrating their own experiences and perspectives into their work, thereby fostering a more meaningful and personalized learning process. Equity was achieved by encouraging students to incorporate elements that reflect their individual interests. Submissions featured an array of creative choices, such as ethnic restaurants, unique book titles, and varied movie selections. This variety highlights the curriculum's success in allowing students to express their cultural backgrounds and personal interests, ensuring every voice is represented and valued.

Despite these positive aspects, student engagement showed a noticeable decline by Day 9, before the introduction of CSS, suggesting that the curriculum's length and structure may still be overwhelming for some. Teacher feedback corroborated this, pointing out the daunting nature of seeing all 25 days of the curriculum at once.

Future Work

In response to the feedback, several enhancements have been made to the ECS-WL tool, pending evaluation in real classroom environments. Teachers can now edit every aspect of the instructions for each day in the curriculum. To make the curriculum more manageable for students, the system hides days that the teacher chooses to keep out of view. Classroom management and personalized instruction have been improved by distinguishing between separate ECS classes taught by the same teacher. Grouping features have been introduced to facilitate collaborative learning. To maintain student engagement, the curriculum is more flexible, allowing for the reduction of days or the addition of extra days for reinforcing content. An overview menu provides a comprehensive view of individual student progress. Lastly, a workspace area enables students to apply daily concepts to their personal website projects, fostering creativity and practical application of skills. The Spanish version of the website is still in the early stages of development and is not available yet.

The updated version ten of the ECS curriculum came out after the pilot of this tool and includes several improvements to increase student engagement in WD. Instead of dedicating an entire unit to WD, it now integrates WD throughout various parts of the curriculum. An updated version of ECS-WL designed to incorporate these changes is currently under development.

Finally, to ensure the tool's long-term success and adaptability, ongoing feedback from both students and teachers should be systematically collected and incorporated into future updates. This iterative approach will enable continuous improvement, ensuring that ECS-WL remains a valuable and effective resource for WD education within the ECS curriculum.

Acknowledgments

This work was funded in part by NSF grant CNS-1923597.

References

- [1] Jane Margolis, Jean J. Ryoo, Cueponcaxochitl D. M. Sandoval, Clifford Lee, Joanna Goode, and Gail Chapman. Beyond access: broadening participation in high school computer science. *ACM Inroads*, 3(4): 72–78, December 2012. ISSN 2153-2184. doi: 10.1145/2381083.2381102. URL <https://doi.org/10.1145/2381083.2381102>.
- [2] Joanna Goode, Jane Margolis, and Gail Chapman. Curriculum is not enough: the educational theory and research foundation of the exploring computer science professional development model. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 493–498, Atlanta Georgia USA, March 2014. ACM. ISBN 9781450326056. doi: 10.1145/2538862.2538948. URL <https://dl.acm.org/doi/10.1145/2538862.2538948>.
- [3] Naomi Blaushild, Raisa Blazquez, Steven McGee, and Randi McGee-Tekula. Investigating How Computer Science Teachers and Instructional Coaches Navigated Remote Professional Development During COVID-19. In *[Paper presentation]*, Denver, CO, 2023. Association For Education Finance & Policy Annual Meeting. URL <https://doi.org/10.51420/conf.2023.1>.
- [4] Corrinne Hess. Wisconsin students missed nearly a month of school last year, May 2023. URL <https://www.wpr.org/education/wisconsin-students-missed-month-school-last-year-absenteeism>.
- [5] Lucia Dettori, Ronald I. Greenberg, Steven McGee, Dale Reed, Brenda Wilkerson, and Don Yanek. CS as a Graduation Requirement: Catalyst for Systemic Change. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 406–407, Baltimore Maryland USA, February 2018. ACM. ISBN 9781450351034. doi: 10.1145/3159450.3159646. URL <https://dl.acm.org/doi/10.1145/3159450.3159646>.
- [6] Chicago Alliance For Equity in Computer Science (CAFÉCS). Impact | CAFÉCS, 2021. URL <https://cafecs.org/impact/>.
- [7] Steven McGee, R. McGee-Tekula, J. Duck, L. Dettori, D. Yanek, A. Rasmussen, Ronald I. Greenberg, and Dale F. Reed. Does exploring computer science increase computer science enrollment? In *Annual Meeting of the American Educational Research Association*, New York, NY, April 2018. American Educational Research Association. URL https://www.academia.edu/36446926/Does_Exploring_Computer_Science_Increase_Computer_Science_Enrollment.
- [8] Joanna Goode, Gail Chapman, and Jane Margolis. Beyond curriculum: the exploring computer science program. *ACM Inroads*, 3(2):47–53, June 2012. ISSN 2153-2184, 2153-2192. doi: 10.1145/2189835.2189851. URL <https://dl.acm.org/doi/10.1145/2189835.2189851>.
- [9] Paul Bruno and Morgan Polikoff. The Effect of Computer Science Course-Taking on Science Outcomes in Chicago Public Schools Effect of CS on Science Grades. Technical report, The Learning Partnership, July 2023. URL <https://jointhepartnership.net/publications/the-effect-of-computer-science-course-taking-on-science-outcomes-in-chicago-public-schools-effect-of-cs-on-science-grades/>.
- [10] Heather Bort, Shion Guha, and Dennis Brylow. The impact of exploring computer science in wisconsin: where disadvantage is an advantage. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE 2018, page 57–62, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450357074. doi: 10.1145/3197091.3197140. URL <https://doi.org/10.1145/3197091.3197140>.
- [11] Matthew Bajzek, Heather Bort, Omokolade Hunpatin, Luke Mivshek, Tyler Much, Casey O’Hare, and Dennis Brylow. MUzECS: Embedded blocks for exploring computer science. In *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, pages 127–132, Atlanta, GA, USA, October 2015. IEEE. ISBN 9781467383677. doi: 10.1109/BLOCKS.2015.7369021. URL <http://ieeexplore.ieee.org/document/7369021/>.
- [12] Deborah A. Fields, Debora Lui, and Yasmin B. Kafai. Teaching Computational Thinking with Electronic Textiles: Modeling Iterative Practices and Supporting Personal Projects in Exploring Computer Science. In Siu-Cheung Kong and Harold Abelson, editors, *Computational Thinking Education*, pages 279–294. Springer

- Singapore, Singapore, 2019. ISBN 9789811365270 9789811365287. doi: 10.1007/978-981-13-6528-7_16. URL http://link.springer.com/10.1007/978-981-13-6528-7_16.
- [13] Chungsoo Na. Assessing Chicago Public School Students' Computer Science Knowledge: Applying a Cognitive Diagnostic Assessment. Technical report, The Learning Partnership, September 2024. URL <https://www.jointhepartnership.net/publications/assessing-chicago-public-school-students-computer-science-knowledge-applying-a-cognitive-diagnostic-assessment/>.
- [14] John Wachen, Steven McGee, Don Yanek, and Valerie Curry. Coaching Teachers of Exploring Computer Science: A Report on Four Years of Implementation. Technical report, The Learning Partnership, January 2021. URL <https://www.jointhepartnership.net/publications/coaching-teachers-of-exploring-computer-science-a-report-on-four-years-of-implementation/>.
- [15] Hauwa Muibi, Brian Dorn, and Thomas Park. Teacher Perspectives on Web Design Instruction. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, pages 231–236, Vilnius Lithuania, June 2015. ACM. ISBN 9781450334402. doi: 10.1145/2729094.2742606. URL <https://dl.acm.org/doi/10.1145/2729094.2742606>.
- [16] Rebecca Mercuri, Nira Herrmann, and Jeffrey Popyack. Using HTML and JavaScript in introductory programming courses. *ACM SIGCSE Bulletin*, 30(1):176–180, March 1998. ISSN 0097-8418. doi: 10.1145/274790.273754. URL <https://dl.acm.org/doi/10.1145/274790.273754>.
- [17] Thomas H. Park and Susan Wiedenbeck. Learning web development: challenges at an earlier stage of computing education. In *Proceedings of the seventh international workshop on Computing education research*, pages 125–132, Providence Rhode Island USA, August 2011. ACM. ISBN 9781450308298. doi: 10.1145/2016911.2016937. URL <https://dl.acm.org/doi/10.1145/2016911.2016937>.
- [18] Thomas H. Park, Ankur Saxena, Swathi Jagannath, Susan Wiedenbeck, and Andrea Forte. OpenHTML: designing a transitional web editor for novices. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, pages 1863–1868, Paris France, April 2013. ACM. ISBN 9781450319522. doi: 10.1145/2468356.2468690. URL <https://dl.acm.org/doi/10.1145/2468356.2468690>.
- [19] Thomas H. Park, Ankur Saxena, Swathi Jagannath, Susan Wiedenbeck, and Andrea Forte. Towards a taxonomy of errors in HTML and CSS. In *Proceedings of the ninth annual international ACM conference on International computing education research*, pages 75–82, San Diego San California USA, August 2013. ACM. ISBN 9781450322430. doi: 10.1145/2493394.2493405. URL <https://dl.acm.org/doi/10.1145/2493394.2493405>.
- [20] Thomas H. Park, Brian Dorn, and Andrea Forte. An Analysis of HTML and CSS Syntax Errors in a Web Development Course. *ACM Transactions on Computing Education*, 15(1):1–21, March 2015. ISSN 1946-6226. doi: 10.1145/2700514. URL <https://dl.acm.org/doi/10.1145/2700514>.
- [21] Refsnes Data AS. W3Schools (HTML), 1998. URL <https://www.w3schools.com/html/>.
- [22] Ron Reiter. Learn HTML - Free Interactive HTML Tutorial, 2017. URL <https://www.learn-html.org>.
- [23] Zach Sims and Ryan Bubinski. Codecademy HTML, 2011. URL <https://www.codecademy.com/learn/learn-html>.
- [24] Chrome DevRel team. web.dev HTML, 2022. URL <https://web.dev/learn/html>.
- [25] Google. Firebase | Google's Mobile and Web App Development Platform, 2010. URL <https://firebase.google.com/>.
- [26] Marijn Haverbeke. CodeMirror 5. <https://github.com/codemirror/codemirror5>, 2017. URL <https://codemirror.net/5/>. Version 5.65.16, Released November 20, 2023.
- [27] Cure53. Dompurify. <https://github.com/cure53/DOMPurify>, 2014. Version 3.1.6.
- [28] Kapor Center. Culturally responsive-sustaining computer science education: A framework, 2021. URL <https://kaporfoundation.org/publications/>.

- [29] OpenAI. ChatGPT, 2024. URL <https://chatgpt.com>.
- [30] Ryan L Boyd, Ashwini Ashokkumar, Sarah Seraj, and James W Pennebaker. The development and psychometric properties of LIWC-22. Technical report, University of Texas at Austin, 2022. URL <https://www.liwc.app/>.
- [31] Matthew L. Newman, James W. Pennebaker, Diane S. Berry, and Jane M. Richards. Lying Words: Predicting Deception from Linguistic Styles. *Personality and Social Psychology Bulletin*, 29(5):665–675, May 2003. ISSN 0146-1672, 1552-7433. doi: 10.1177/0146167203029005010. URL <http://journals.sagepub.com/doi/10.1177/0146167203029005010>.
- [32] Seth C. Kalichman and Joshua M. Smyth. “And you don’t like, don’t like the way I talk”: Authenticity in the language of bruce springsteen. *Psychology of Aesthetics, Creativity, and the Arts*, 17(5):581–589, October 2023. ISSN 1931-390X, 1931-3896. doi: 10.1037/aca0000402. URL <https://doi.apa.org/doi/10.1037/aca0000402>.