WebTrainingWheels: A Platform for Interactively Learning Web-Based Workflows

Peifeng He*
Duke University
Durham, USA
ph162@duke.edu

Ziheng Wang*
UNC - Chapel Hill
Chapel Hill, USA
wzh@unc.edu

Prasun Dewan
UNC - Chapel Hill
Chapel Hill, USA
dewan@cs.unc.edu

***Abstract*—As desktop applications and shared websites evolve and become more complex, potentially outdated traditional text, image, and video-based tutorials often struggle to meet user needs. To address these limitations, interactive tutorials have emerged, with prior efforts focusing on hosting them for desktop applications. As modern web interfaces become complex, there is a growing need for platforms to support interactive tutorials for these interfaces as well. In this paper, we introduce WEBTRAININGWHEELS (WTW), a Chrome-based platform that enables users to *create, retrieve, follow, and edit interactive visual tutorials to execute a workflow originating from any website*. A tutorial is created by demonstrating the series of web actions needed to accomplish it. Upon entering a website that forms the starting point for one or more tutorials, users can select and follow a tutorial step-by-step. Through user studies (n=8), we demonstrate that our platform improves efficiency and task completion rates, and that participants prefer WTW for unfamiliar tasks. We further develop an *AI-to-demonstration* system that uses large language models (LLMs) to generate tutorials on the fly, based on user-defined goals.**

***Index Terms*—web platform, service innovation, interactive tutorials, collaborative learning, AI agent**

I. Introduction

As applications and websites grow in sophistication, their user interfaces (UI) often become more intricate, requiring users to navigate complex structures, settings, and interaction sequences [1]–[3]. In order to ensure that users can efficiently learn how to interact with UI elements across different systems, tutorials have been created for shared websites and local desktop applications [4], [5]. Traditional tutorials include text-based guides with auxiliary videos and images. However, such tutorials have many disadvantages, including being static and lacking support for updates [3], [4], requiring users to switch between contexts [6], and presenting complex textual descriptions that hinder task completion [7], [8].

To address the challenges posed by traditional tutorials, researchers have begun to leverage graphical annotations, spatial cues, and direct interface highlighting to communicate with users [9]–[11]. This shift has led to the development of interactive visual tutorials, where the system visually marks the UI elements involved in each instructional step [9], [12]. Primitive interactive visual tutorials were typically embedded directly into the source code [7], [13]. While functional, these tutorials are limited in scope—they guided users through only predefined tasks, and their creation was restricted to platform developers with access to the source code. To address these limitations, researchers introduced interactive visual tutorial platforms, allowing users to create, share, and use tutorials without access to source code [14]–[16]. The tutorials can now be created by arbitrary users – not just the developers. Previous platforms have focused on desktop applications [16], system operations [15], and mobile applications [14].

With the growing complexity of modern web applications and browser-based workflows [2], [17], [18], there is a clear need for platforms that support interactive tutorials in web environments. Modern web user interfaces, built on structured and accessible *Document Object Models (DOMs)*, offer new opportunities for interactive visual tutorials. Building on these concerns and foundations, we introduce WEBTRAININGWHEELS (WTW)—a demonstration-based open interactive visual tutorial platform for the Chrome browser.

In Creating Mode, users can create tutorials through user demonstration on any website. For each step they want the tutorial user to take, they highlight the corresponding UI element on the webpage by clicking it and provide the instructions they want the user to see. In Menu Mode, if a website has at least one WTW tutorial, a pop-up window emerges and displays all the workflows available on that URL. In Following Mode, users click on the highlighted UI elements on the current page to proceed to the next step. Our design also includes the downloading and editing of the tutorials to facilitate sharing and updating.

Recent advances in web-oriented AI agents—such as Anthropic's Computer Use [19] and OpenAI's Operator [20]—show that modern vision/language models can perceive and manipulate arbitrary graphical interfaces with high accuracy. Within the Chrome ecosystem, open-source extensions including A5-Browser-Use [21], NanoBrowser [22], and Midscene.js [23] bring such automation directly into the browser. Yet all of these systems focus on autonomously completing tasks; none of them guide a human step-by-step through a workflow needed to complete a task. To investigate how agent-based perception can support human learning, we extend WTW with an *AI-to-demonstration mode* that lets large language models interpret web UIs and automatically generate interactive visual tutorials from user-defined goals, thus uniting AI autonomous execution with demonstration-style instruction.

Compared to existing non-web-based platforms [14]–[16], WTW's advancements include eliminating the need for screenshots or screen recordings to create tutorials, removing the step of retrieving and downloading tutorials, enabling in-context tutorial use, and supporting updates to existing

*Equal contribution.

tutorials. Our user study indicates that, compared to traditional tutorials, participants prefer to choose WTW for unfamiliar tasks. For the same task, users finished the task faster and with higher accuracy using WTW.

Our contributions in this paper are as follows:

- We introduce **WebTrainingWheels** (WTW), a Chrome extension that enables end-users to produce, fetch, edit, and follow interactive visual tutorials starting from any website, demonstrating how guidance and support services are created and accessed on the web.
- We extend WTW with an AI-to-demonstration mode that employs LLMs to automatically synthesize interactive tutorials from natural-language goals.
- We conducted a user study that demonstrates WTW improves users' efficiency, task completion rate, and overall experience when learning unfamiliar workflows, highlighting its value as an effective platform for web-based workflow learning.

II. Related works

A. Creating Visual Tutorials

Existing works propose two main approaches for creating visual tutorials: either with [4], [5], [24] or without human participation. HelpViz [14] collects text-based tutorials and uses a deep language model to convert them into visual tutorials. The ones with human participation include two main methods: user demonstration [5], [15], [16] and crowdsourcing [25], [26]. Crowdsourcing involves the collaborative creation or annotation of tasks by multiple creators. Our goal is to allow users to create complete visual tutorials using a visual interface, making user demonstration the ideal choice. In previous works, with the help of loggers and monitoring tools, the user demonstration process captures the creator's operations by using screen recordings or screenshots, and generates each step for the creator to edit [15], [16]. However, in this process, there is a chance that the system will not recognize the UI element that should be clicked, and an error handling pipeline needs to be designed [16]. In our approach, we eliminate the need for video or screenshot-based recording. Instead, we identify each action by capturing the exact location of the UI element selected by the creator within the DOM tree, which ensures that all tutorials work perfectly according to the creator's specification.

B. Following Visual Tutorials

In existing works, users often need multiple screens to follow a tutorial. HelpViz provides guidance by requiring users to switch to a separate tutorial interface [14]; Torta delivers mixed-media system-level tutorials within a web browser [15]. In WTW, we want the user to be able to finish the task in place, without leaving the task interface.

For an interactive visual tutorial platform, it is crucial to determine whether the user correctly follows the tutorial instructions. In order to receive and check user interactions, existing app-based tutorial systems gather details of user actions using accessibility APIs to obtain the system's accessibility data [14]–[16]. In WTW, we gain information about user actions by setting up a click event listener.

C. AI Web Agents

A large amount of recent research focused on developing AI agents, which are LLM powered tools that automate tasks. General AI agents, also known as computer using agents [19], [27], aim to automate any task the user may have given as a natural language prompt. Examples include Anthropic's Computer Use [19] and OpenAI's Operator [20]. A subcategory of such agents are browser using agents, which focus on automating tasks within the browser environment [21]–[23], [28].

Agent implementation can roughly be divided, based on the AI backbone they use, into vision language model (VLM) or natural language model (NLM) based agents. WTW adopts a natural language model (NLM)–based framework. NLM models first index the screen into interactive elements and give the elements information as texts to a pretrained language model [28].

All agents are designed to be autonomous. Thus, users cannot follow along with or learn a task from the agent. However, we still aim for users to be actively involved in the task completion process, so that when they encounter the same task again, they are able to accomplish it on their own.

III. System Design

Interactive Visual Tutorial Our tutorials are interactive visual tutorials - the users are guided to complete the task within the web interface required for the task, without the need for any other interfaces.

No Contribution Barrier Anyone can be a creator of a tutorial. After installing WTW as a Chrome extension, any user can define the website(s) that their tutorial will serve, the name and description of the tutorial, the UI elements that will be clicked, and the types of actions to be performed.

Generalizing to Website Families WTW operates on any website, which aligns with analogous design principles in app-based or mobile-based platforms [16]. A tutorial can start and end at any URL and must contain at least one step. To support reusability, tutorials should function seamlessly across webpages that share the same page structure, regardless of user-specific paths or session parameters. For example, if we have a tutorial on a webpage with URL *gradescope.com/course/<courseId>*, this tutorial is available in all different course websites represented by different *courseId*.

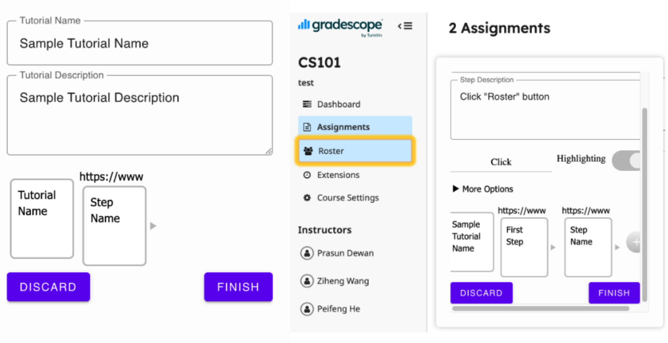
Easy to Access WTW eliminates the need for users to search for and download tutorials. By simply going to a website, the user can find all tutorials that start from the current webpage. We also support partial tutorials (i.e., users can start from the middle of a tutorial).

Editing and Uploading WTW also supports editing, downloading, and uploading of tutorials after creation. Users can update the step names and step instructions. Additionally, WTW is able to address changes in the relative location of an element when the webpage's UI is updated.

Minimal Interruptions To minimize interruptions to the user's normal web experience, the WTW shows Chrome popups only when there are tutorials available. All popups are draggable. Users can disable interfaces they consider non-essential.

IV. System Implementation

WTW can be partitioned based on three modes it supports: Creating Mode, Menu Mode, and Following Mode. Each mode represents a system state and is checked and recorded upon entering a new webpage.



Tutorial Name

Sample Tutorial Name

Tutorial Description

Sample Tutorial Description

Step Description

Click "Roster" button

Fig. 1. Tutorial creation. On the left, we show the panel embedded on the webpage when the user starts building a tutorial. Below the name and description fields is the steps section. Each block represents an individual step in the tutorial, except for the first one, which displays the tutorial name. When the user clicks a step node, the panel changes to allow the user to edit the specific step (right). We allow specifying a name, description, highlighted UI, action type (here "click"), and URL. The "Highlighting" toggle allows the user to demonstrate clicking while not triggering the clicked element's event listeners, which may include redirecting.

A. Creating Mode

Users can bring up a tutorial creation panel on *any* website. As a Chrome extension, all modules are embedded within web pages. Users can edit the tutorial's name, description, and details of steps all within the webpage they are viewing (Figure 1, left).

Within each step, the creator first needs to enter the name of the step and the text instruction. Users can then highlight an element on the screen, which corresponds to the action required for this step (Figure 1, right). After a step is created, the creator has two choices. They can finish the tutorial by clicking the "finish" button. The alternative is to click the add-step button to the right of the current step container. When this button is clicked, our system first simulates a click on the highlighted UI element and then refreshes the panel in preparation for adding a new step. The steps can span multiple websites associated with different domains to allow arbitrary web-based workflows.

Once a tutorial is completed, we collect two sets of information and store them in the database: the tutorial collection, which stores all tutorial objects—each identified by a unique ID—and the domain collection, which contains domain-related data used for lookup and organization.

The domain collection comprises multiple documents, each named after a host domain (e.g., *gradescope.com, google.com.*) that contains at least one tutorial. If a tutorial contains steps across multiple domains, it appears under multiple documents. Each document in the domain collection contains all tutorial IDs corresponding to tutorials hosted under that domain, and each ID includes a list of visited URLs represented in regular expression form. Our regular expression transformation involves segmenting the URL based on path delimiters. All purely numeric components other than the domain are replaced with a symbol representing arbitrary content in the regular expression. For example, if a tutorial visited *gradescope.com/courses/328434*, WTW transforms this URL into *gradescope.com/courses/** and then records it.

B. Menu Mode

In Menu Mode, all available tutorials for the current webpage are displayed to the user upon entering the webpage (Figure 2, Menu Selection).

First, WTW matches the domain of the current webpage with the domains in the domain collection. If the current domain is not in the domain collection, then no WTW tutorial has been created in that domain, and nothing will pop up.

If the domain is matched, we iterate through all the tutorials under that domain. If a tutorial's URL list contains a URL (in the regular expression form introduced in section IV-A) that matches the current URL (also transformed to the regular expression form), WTW records the ID of the tutorial, which we can then retrieve from the tutorial collection where the tutorial objects are stored. For example, suppose a tutorial was created on *gradescope.com/courses/328434*. Later, when a user visits another course page such as *gradescope.com/courses/987492*, WTW applies the same regular expression transformation to match against stored URLs for the tutorial originally created for course "*328434*". Since both URLs can be transformed to the same regular expression pattern, the tutorial created for the course "*328434*" will be made available for the course "*987492*" as well.

C. Following Mode

By selecting a tutorial in the tutorial menu popup (Figure 2, Menu Selection), WTW goes into the Following Mode, guiding the user through the task via visual highlighting.

During the following process, there are two sections on the webpage: an updated menu bar and a highlighted UI element with text instructions next to it.

The menu bar becomes an ordered list of steps to indicate the user's current location and the relative location in the tutorial. On the page, WTW highlights the UI element required for the current step. Our menu bar is always draggable and can be closed, thus avoiding covering the highlighted UI element. The corresponding instruction is displayed next to the element, with an option for the user to hide the instruction box (Figure 2). The user completes the step by clicking on or typing in the highlighted UI element, after which WTW automatically jumps to the next step. If the user performs an incorrect operation and, after its completion, WTW cannot locate the UI element that should have been highlighted in that step, it indicates that the user has navigated to the wrong webpage due to the mistake and prompts them to return to the previous page to continue the task.

When a user clicks on a highlighted UI element that triggers a redirection to a new URL, the system checks the current status. Since the system is in the Following Mode, the tutorial searching and matching mechanism is not triggered, as it is only active in the Menu Mode. WTW highlights the UI element for the next step, and the user is not interrupted.

After the user completes a task, the step list in the menu bar disappears and changes back to the Menu Mode, so that the user can select the next tutorial if there exists one.

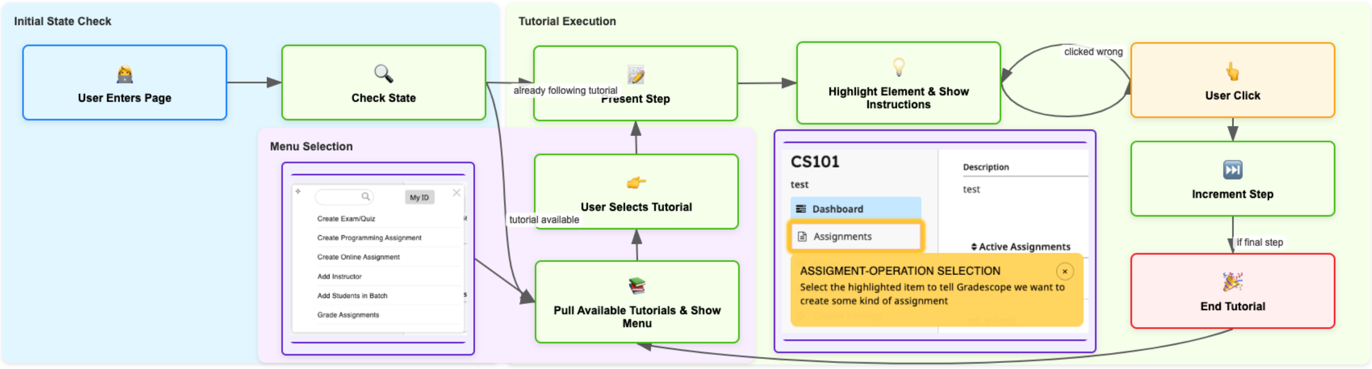
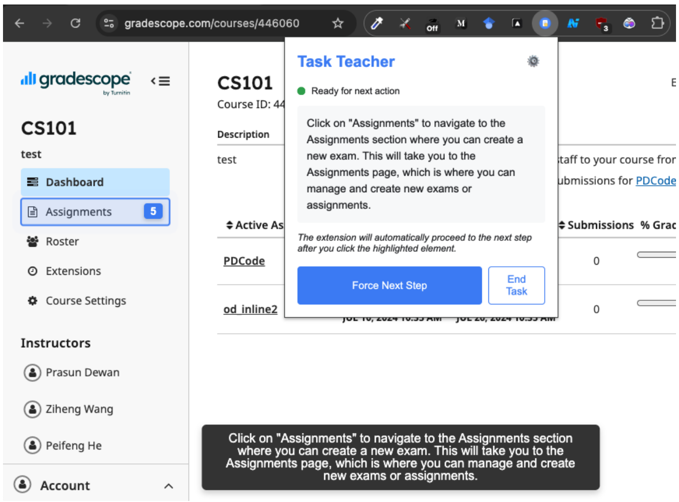


Fig. 2. Workflow of tutorial following. Upon entering any webpage, our extension checks the stored states and decides whether to show a tutorial menu or resume a running tutorial, or stay idle if neither is available. During tutorial execution, the extension loads the current step and highlights the corresponding UI element. If the user clicks on this element, we increment the step and start to present the next step; if not, we use various safety mechanisms to guide the user and re-highlight the UI. After completing the final step, we automatically load a new tutorials menu on that finish page.

AI-to-demonstration Mode We extend WTW with the ability to incorporate language models during workflow learning (Figure 3). The model reasons over the prompt and the screen elements to predict the index of the UI element most likely to lead to task completion. WTW then highlights this element for the user to click. One popular NLM based browser agent is *browser-use* [28], on which we based our AI WTW.

Under this mode, no pre-defined tutorials are required. The system dynamically interprets user prompts and provides guidance directly based on model predictions. Therefore, AI WTW only provides the Following Mode experience with no editing or creating of tutorials.



Ready for next action

Click on "Assignments" to navigate to the Assignments section where you can create a new exam. This will take you to the Assignments page, which is where you can manage and create new exams or assignments.

The extension will automatically proceed to the next step after you click the highlighted element.

Click on "Assignments" to navigate to the Assignments section where you can create a new exam. This will take you to the Assignments page, which is where you can manage and create new exams or assignments.

Fig. 3. Our experimental AI-to-demonstration solution. Users enter a task prompt, and the extension will analyze the webpage with this prompt and highlight the element most likely to lead to completion of the task.

V. System Management

A. Storage and Privacy

Our data is stored in an online database called *Firebase* [29]. The two main blocks of data stored are all the tutorial objects and all the domain names that contain tutorial(s), as introduced in the previous section.

During tutorial creation, WTW records only the DOM location of clicked elements. During tutorial fetching and execution, no user data or private information is transmitted. WTW ensures no exposure of private data.

B. Editing

The user can click on the update button of the tutorial in the Menu Mode, which will take the user back to the Creating Mode. The user can change the name and description of each step in the panel of the Creating Mode. We did not add the operation of deleting or adding a step because most of the tutorials involve redirecting to a new URL, and the operation of modifying step structures may cause all the steps after it to be disconnected from the tutorial. In such cases, the amount of work required to edit steps is not much different from recreating the tutorial.

But we found that sometimes the website will update its own interface, resulting in the position of some UI elements in the DOM tree to be changed. For example, the UI element's position in the DOM tree may change from the third child of *<body>* to the fourth. In this case, users can download the tutorial object to get a JSON file, and they can adjust the parameters of the UI location in a certain step in the JSON file. After the modification is done, the JSON file can be uploaded, and WTW can finish creating the updated tutorial according to this JSON file. As administrators, developers can also modify the parameters of these UI locations directly in the database.

VI. User Study

In order to test the usefulness of WTW, a user study was designed in which each participant was required to complete three tasks on *gradescope.com*, which is an online assessment platform that supports submitting, grading, and managing multiple types of assignments.

- Task A: Create a multiple-choice online assignment in Gradescope. (14 steps)
- Task B: Create an exam/quiz using a PDF template in Gradescope. (16 steps)
- Wrap-up Task: Create a programming assignment with a custom-written autograder in Gradescope. (14 steps)

Participants were randomly divided into two groups: Group 1 and Group 2. Group 1 did Task A with WTW and Task B using traditional web resources *provided by Gradescope* (text-based instructions with auxiliary images), while Group 2 did Task A using traditional resources and Task B with WTW. We ensured that both the traditional and WTW tutorials conveyed precisely the same workflow by constructing the WTW tutorial from the traditional counterpart.

For the wrap-up task, participants had the option to choose between the traditional methods and WTW. After they had done all the tasks, they were asked to complete a survey. They also submitted a screen recording of performing Tasks A, B, and the wrap-up task. We received a total of eight responses from eight participants (P1-P8). Four participants were in Group 1, and the other four participants were in Group 2. All eight participants were college students between the ages of 20-25, and none of them had completed these tasks before the study.

A. Finishing Time

We measured the time each group took to complete each task and calculated the average, as shown in Table I. For users who used WTW, the start time was recorded when they clicked on the corresponding tutorial. For those using the traditional tutorials, the start time was when they opened the tutorial's web link. In both cases, the end time was marked when the participant clicked the final button of the task.

Our results show that WTW improved participants' task completion speed. For Task B, since some users missed steps when using the traditional tutorial (explained in Section C), we excluded the corresponding steps from the WTW counterpart for a fair comparison.

TABLE I
AVERAGE FINISHING TIME FOR TASK A AND B

	Finishing Time (Seconds)	
Task	*With WTW Tutorial*	*With Traditional Tutorial*
Task A	162.5	209.75
Task B	104	143.75

B. User preference

We asked all participants to choose the method they want for the wrap-up task. When faced with an unknown task, we wanted to know if participants are willing to learn it with WTW or a traditional tutorial.

In the study, participants were provided with a direct link to the traditional tutorial, eliminating the process of searching and retrieval. Despite this convenience, all participants chose to use WTW to complete the wrap-up task. We also collected the reasons behind participants' choices. P1 said WTW was more "convenient". P3 said WTW was "smoother" and more time-efficient. P5 said WTW did not require switching interfaces. P6 said WTW "specifies what I should do to save my time".

C. Task Completeness

For Task A, everyone completed the task correctly. However, in Task B, all participants using the traditional Gradescope tutorial missed the step to create an ID region, a name region, and a question region. One possible reason is that, while the image initially displayed these three regions, participants may not have remembered to take the actions suggested by the image after completing the subsequent text-based steps. This caused the participants to ignore these steps. We use this real-world example to illustrate that traditional tutorials cannot guarantee accurate task completion, whereas WTW ensures correctness through its design. At the same time, we aim to separate the influence of tutorial quality from users' preferences and feedback on WTW, as our goal is to highlight the advantages of the platform's design feature rather than the content itself. To this end, users did not receive any feedback regarding the accuracy of their task performance, ensuring that their choices and evaluations of WTW were made independently of its demonstrated accuracy.

D. Usability

We have collected participants' opinions on the usability of WTW. First, we asked users whether they liked the fact that they did not need to switch windows, but instead completed all tasks in one screen. All users agreed with this advantage. However, we were concerned that overlaying all the information might lead to a cluttered webpage. So we asked the participants if they felt the interactive visual tutorial made the webpage feel overcrowded. One participant stayed neutral, while the rest agreed that the existence of WTW did not disrupt their online experience. We also collected information on whether the participants believed that WTW improves efficiency and is less error prone. All participants chose to agree except one participant who chose to be neutral in the efficiency question.

VII. Discussion and Future Work

WTW enables users not only to follow interactive web tutorials but also to create and share them with others, offering a new model of user-driven guidance that is both accessible and adaptable. By lowering the barrier to content creation and embedding assistance directly within task environments, WTW supports more effective web-based workflow learning.

System Maintenance A drawback of our current system is difficulties in maintaining tutorials. Creating tutorials requires task-specific expertise and is labor-intensive, as the creator needs to demonstrate and label each step. After creation, a tutorial stores fixed information used to locate the highlighted elements of the tutorial steps. However, websites may change their UI layouts, making the current UI locating system dysfunctional. Although we can manually update the UI location for each step, these challenges are well suited for AI and our AI-to-demonstration mode suggests that replacing the manual creation module with an LLM may address the problems.

URL Matching Algorithm For a given current URL, WTW uses regular expressions to match it with stored URLs. We treat purely numeric route parameters as dynamic identifiers that can be generalized. However, when a segment becomes alphanumeric, it's usually unclear whether it is dynamic or fixed (so we shouldn't generalize it). To avoid overgeneralization, WTW only transforms purely numeric segments into regular expressions and treats others as fixed.

Environment Restriction WTW only works on Chrome browser now. Our next step will be to refine our experimental

AI system and develop a more robust framework across different browsers.

VIII. Conclusion

We present WebTrainingWheels, an online visual tutorial platform for web-based workflow learning. By lowering the barrier to both tutorial creation and consumption, WTW empowers users to contribute to a shared knowledge platform. Through a user study, we demonstrate that users prefer WTW over traditional tutorials when completing unfamiliar tasks. Compared to traditional tutorials, WTW enhances user productivity, improves task completion accuracy, and supports better user focus by eliminating context switching and minimizing disruption.

To the best of our knowledge, WTW is the first demonstration-based interactive tutorial platform for the browser. Unlike prior platforms, we eliminate the need to: (a) rely on recorded materials (screenshots and recordings) and system accessibility APIs, (b) require users to switch between contexts to accomplish tasks, (c) need to manually retrieve and load tutorials, and (d) account for potential errors in the process of creating the tutorial. In addition, the extended AI WTW bridges the gap between autonomous task execution and human-centered learning. This hybrid design opens new directions for collaborative systems that both teach and assist.

Acknowledgment

This work was funded in part by NSF award OAC 1829752. The user study was performed with the approval of University's IRB. The consent from the human subjects in the research was obtained.

References

[1] J. E. Yu and D. Chattopadhyay, "Reducing the search space on demand helps older adults find mobile ui features quickly, on par with younger adults," in *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. New York, NY, USA: ACM, 2024, p. 22.

[2] A. Miniukovich, S. Sulpizio, and A. D. Angeli, "Visual complexity of graphical user interfaces," in *Proceedings of the 2018 International Working Conference on Advanced Visual Interfaces (AVI '18)*, 2018, p. 9.

[3] M. Han and P. Park, "A study of interface design for widgets in web services through usability evaluation," in *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human (ICIS '09)*. New York, NY, USA: Association for Computing Machinery, 2009, pp. 1013–1018. [Online]. Available: https://doi.org/10.1145/1655925.1656109

[4] P. Chi, S. Ahn, A. Ren, M. Dontcheva, W. Li, and B. Hartmann, "Mixt: Automatic generation of step-by-step mixed media tutorials," in *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*, 2012, pp. 93–102.

[5] F. Grabler, M. Agrawala, W. Li, M. Dontcheva, and T. Igarashi, "Generating photo manipulation tutorials by demonstration," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, p. 66, 2009.

[6] H. Cai, "Facilitating information management in integrated development environments through visual interface enhancements," in *2015 IEEE International Conference on Software Quality, Reliability and Security - Companion*. IEEE, Aug. 2015, p. 221–229. [Online]. Available: http://dx.doi.org/10.1109/QRS-C.2015.46

[7] C. Kelleher and R. Pausch, "Stencils-based tutorials: Design and evaluation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*, 2005, pp. 541–550.

[8] T. Jong, "Cognitive load theory, educational research, and instructional design: Some food for thought," https://doi.org/10.1007/s11251-009-9110-0, 2010.

[9] Jamet and J. Fernandez, "Enhancing interactive tutorial effectiveness through visual cueing," *Educational Technology Research and Development*, vol. 64, pp. 631–641, 2016.

[10] J. Heer and B. Shneiderman, "Interactive dynamics for visual analysis," *Communications of the ACM*, vol. 55, no. 4, pp. 45–54, 2012. [Online]. Available: https://doi.org/10.1145/2133806.2133821

[11] C.-Y. Wang, W.-C. Chu, H.-R. Chen, C.-Y. Hsu, and M. Y. Chen, "Evertutor: Automatically creating interactive guided tutorials on smartphones by user demonstration," https://doi.org/10.1145/2556288.2557407, 2014.

[12] T. Yeh, T. Chang, B. Xie, G. Walsh, I. Watkins, K. Wongsuphasawat, M. Huang, L. Davis, and B. Bederson, "Creating contextual help for guis using screenshots," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*, 2011, pp. 145–154.

[13] A. Cypher, D. C. Halbert, D. Kurlander, H. Lieberman, D. Maulsby, B. A. Myers, and A. Turransky, Eds., *Watch What I Do: Programming by Demonstration*. Cambridge, MA, USA: MIT Press, 1993.

[14] M. Zhong, G. Li, P. Chi, and Y. Li, "HelpViz: Automatic generation of contextual visual mobile tutorials from text-based instructions," in *Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 1144–1153. [Online]. Available: https://doi.org/10.1145/3472749.3474812

[15] A. Mysore and P. J. Guo, "Torta: Generating mixed-media gui and command-line app tutorials using operating-system-wide activity tracing," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 703–714. [Online]. Available: https://doi.org/10.1145/3126594.3126628

[16] G. Li, T. Lu, J. Yang, X. Zhou, X. Ding, and N. Gu, "Intelligently creating contextual tutorials for gui applications," in *2015 IEEE 12th International Conference on Ubiquitous Intelligence and Computing and 2015 IEEE 12th International Conference on Autonomic and Trusted Computing and 2015 IEEE 15th International Conference on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*. Beijing, China: IEEE, 2015, pp. 187–196. [Online]. Available: https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.50

[17] M. Butkiewicz, H. V. Madhyastha, and V. Sekar, "Understanding website complexity: Measurements, metrics, and implications," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference (IMC '11)*. New York, NY, USA: Association for Computing Machinery, 2011, pp. 313–328. [Online]. Available: https://doi.org/10.1145/2068816.2068846

[18] V. Panwar, "Web evolution to revolution: Navigating the future of web application development," *International Journal of Computer Trends and Technology*, vol. 72, pp. 34–40, 02 2024.

[19] Anthropic, "Introducing computer use, a new claude 3.5 sonnet, and claude 3.5 haiku," https://www.anthropic.com/news/3-5-models-and-computer-use, 2024.

[20] OpenAI, "Introducing operator," https://openai.com/index/introducing-operator/, 2025.

[21] AgenticA5, "A5-browser-use: Chrome extension and server for agentic browser automation," https://github.com/AgenticA5/A5-Browser-Use, 2025.

[22] Nanobrowser, "nanobrowser/nanobrowser: Open-source chrome extension for ai web automation," https://github.com/nanobrowser/nanobrowser, 2025.

[23] web-infra dev, "web-infra-dev/midscene: Your ai operator for web, android, and more," https://github.com/web-infra-dev/midscene, 2025.

[24] D. Eckhoff, C. Sandor, D. Kalkoten, U. Eck, C. Lins, and A. Hein, "Tutar: Semi-automatic generation of augmented reality tutorials for medical education," in *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 2018, pp. 430–431.

[25] J. Kim, P. T. Nguyen, S. Weir, P. J. Guo, R. C. Miller, and K. Z. Gajos, "Crowdsourcing step-by-step information extraction to enhance existing how-to videos," https://doi.org/10.1145/2556288.2556986, 2014.

[26] M. Gordon and P. Guo, "Codepourri: Creating visual coding tutorials using a volunteer crowd of learners," https://doi.org/10.1109/VLHCC.2015.7357193, 2015.

[27] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J. Wen, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, Mar. 2024. [Online]. Available: http://dx.doi.org/10.1007/s11704-024-40231-1

[28] M. Müller and G. Žunič, "Browser use: Enable ai to control your browser," 2024. [Online]. Available: https://github.com/browser-use/browser-use

[29] G. LLC, "Firebase: Real-time database and backend as a service," https://firebase.google.com, 2025.