

Curriculum Caddy: Enabling Teachers Through Curriculum-Centric AI

Gabrielle (Gabbi) Forsythe
gforsythe6@gatech.edu

Abstract—Novice computer science (CS) teachers often do not have the technical experience to teach CS effectively. They can build these skills through professional development courses; however, these courses often lack support in the long run making it difficult for teachers to continue to build their skills once the course is completed. Curriculum Caddy is a curriculum-centric Generative AI tool that acts as a teaching assistant to teachers and as a tutor to students. This tool is different from other Generative AI tools in that the teacher can upload class related documents (such as curriculum outlines or lesson plans) that are summarized and sent to the model. This enables the model to generate more targeted responses based on what is being taught in the classroom. Currently, this tool is just a proof of concept so no research has been done on its effectiveness, but this tool could have several potential benefits for both students and teachers.

1 INTRODUCTION

Computer science education (CSE) is becoming increasingly embedded in K-12 schools around the globe. In several countries outside of the US, such as Israel, this push for CSE came many years ago (Armoni & Gal-Ezer, 2014). However, the US is just now catching up. This rush to implement computer science (CS) at the K-12 level has made it difficult to find teachers with the proper experience to teach the classes. Because of this shortage, teachers who often specialize in other subjects take on CS classes to teach. This means that novice CS teachers often do not have enough computer science experience to teach effectively at the K-12 level. Teachers can be enabled through professional development courses or through their own research, but often times teachers do not have the time to effectively teach themselves these concepts. They also are lacking a place to go with any one-off questions about topics in the curriculum that may arise throughout the year.

When given the necessary skills and knowledge through professional development courses, teachers consistently feel more confident in the material they are presenting and are more willing to adopt CS courses into their existing courses. Ongoing professional development not only shows benefits for the teachers but also for students. Increasing the number of teachers who are equipped to teach CS leads to increased accessibility of CS courses. These courses drive greater interest in CS which ultimately leads to a more diverse CS field in the future. While teachers have consistently rated their professional development courses for CS highly, they also have identified several areas that could use more training and enablement (Rich et al., 2021). This means that while the work being done on professional development for CS courses has been effective, there is still work to be done to fully enable and equip novice CS teachers.

Curriculum Caddy is a web-based tool that allows students and teachers to chat with a Generative AI tutor (named “Ada”) that has context of the curriculum being used in the classroom. This means that the interactions between the user and the AI tutor are more targeted since they are set within the context of where the class is currently in their learning journey. This tool is seeking to address the professional development and computer science gaps that exist in education today.

2 RELATED WORK

2.1 Existing Professional Development

Current CS teachers typically either have a mastery of CS but little classroom experience or they are great in the classroom but have no CS experience (Goode et al., 2012). Several studies done on computer science curriculums have pointed to professional development and teacher enablement as being one of the most important aspects of a successful computer science program (Fagerlund et al., 2020; Hubwieser et al., 2014; Hubwieser et al., 2015; Lee and Perret, 2022). Huang et al. (2022) and Rich et al. (2021) both developed professional development for teachers in tandem with the CS curriculums that they were developing. The teachers in these studies indicated that this professional development had an extremely positive impact on their ability to teach CS classes effectively. Training the teachers and equipping them to teach Computer Science is a vital part of the process of implementing CS within schools. While these courses are effective,

they often lack ongoing support and ways for teachers to ask questions. Some curriculums and professional development organizers provide online ongoing support where teachers can ask questions about the content or curriculum. However, while many professional development organizers offer this ongoing support, many teachers have indicated that they have not found the online support very helpful or responsive to their inquiries (Friend et al., 2022). Teachers are lacking both time and access to resources that can equip them to teach CS resulting in lower teacher and student satisfaction and less effective CS courses.

2.2 Conversational AI Tutors

Intelligent and Conversational AI tutors generally have shown positive results in terms of the students learning outcomes and performance (Ritter et al., 2004). Research also shows that the impacts of good implementations of these tools can have huge impacts and help address shortages in teaching staff (Conati, 2009). However, these tools can also bring about immense technical challenges such as implementing an effective model of student and tutor behavior (Conati, 2009), as well as ethical concerns like trust, transparency, and more (Ruane et al., 2019).

2.3 ChatGPT for Tutoring

Currently, one of the main uses of ChatGPT in schools is for tutoring students. Studies overwhelmingly agree that the use of ChatGPT brings huge benefits (Deriba et al., 2023). This innovation in AI has essentially allowed anyone with an internet connection to have access to a personal tutor on any topic at any time (Limo et al., 2023). These benefits do not come without challenges. First, there are many ethical considerations to consider, including the fact that some students submit answers directly from ChatGPT which is plagiarism. Additionally, students can become over reliant on the tool, leading to a lack of critical thinking skills (Deriba et al., 2023).

ChatGPT does currently offer similar features to that of Curriculum Caddy in that it allows you to upload documents and create a custom GPT that can be accessed with a unique link. While this would likely accomplish the same functionality, it is not intuitive to teachers that they would have the ability to do this for their classroom. Additionally, a paid subscription to ChatGPT is required for the teacher and in some cases the students that would access the tool. Curriculum

Caddy is designed for the classroom and has classroom management features that are not available with ChatGPT's "Create a GPT" functionality.

3 SOLUTION

3.1 Tool Overview

Curriculum Caddy is a web-based tool that was built using React (for the frontend), Firebase (for the database), and Netlify (for hosting). In its current form, the tool is a proof-of-concept. For example, it doesn't have any login capabilities and the user can flip between student and teacher mode at will. However, these capabilities could easily be added so that the tool can be rolled out in a formal manner.

3.2 Functionality

Curriculum Caddy has three main components: the "Ask Ada" page, the "Curriculum" page, and the "Settings" page. *Ask Ada* is the main page of the tool where the user is able to converse with the Generative AI tutor. Teachers can ask questions about different concepts they are struggling to understand, troubleshoot code, ask for help lesson planning, and more. Similarly, students can interact with Ada the way they would a regular tutor or Generative AI chat tool and ask questions on concepts or topics, troubleshoot code, be quizzed on lesson material, etc.

The *Curriculum* page is the differentiator that sets this tool apart from other AI tutoring tools or Generative AI tools. This is where Teachers can upload documents related to their classroom such as lesson plans, assignment instructions, and curriculum documents. Once these documents are uploaded, they are summarized using Generative AI (see *Section 3.3*) and this is used to provide additional context when the user prompts the AI tutor on the *Ask Ada* page.

Teachers can provide additional details to the tool about themselves and their classroom on the *Settings* page. Here teachers can enter details about the subject of their class, their grade level, and preferred programming language. Additionally, both students and teachers can enter in their name on this page, which the model uses to personalize the interactions it has with the user.

Curriculum Caddy also includes a *Start Here* page where users who are just getting started with the tool will go. This page contains a video user guide that will direct the user on how best to make use of the tool. The user will either be given the Teacher User Guide or the Student User Guide based on their role. Both guides contain a walkthrough of how to use the tool to ask questions, adjust your settings, and example prompts to give on the *Ask Ada* page. The Teacher Guide additionally contains details around how to upload and edit curriculum within the tool and edit the classroom-wide settings.

3.3 Generative AI Capabilities

3.3.1 Generative AI Overview

Curriculum Caddy uses the Gemini API (*Google AI for Developers*) to generate tutor response and to summarize the curriculum the teachers upload. This API works similarly to any other chat-style Generative AI tools where the user (or program) submits a prompt (question or task) to the API, and it returns a text response to that prompt. However, unlike the typical chat-style interface, none of the history of the conversation is retained. For that reason, with every prompt the user inputs, any additional information or context must also be passed to the model in order to get the targeted response that the user is looking for.

3.3.2 Prompt Structure

With each message that the user sends to the tutor, the model actually receives a prompt containing a few different components. First is the Initial Prompt which tells the model about the situation in general. For students, the initial prompt tells the LLM to behave as a tutor and contains guardrails to prevent the model from providing the student with content that contains age-appropriate examples, grade level specific explanations, redirects off topic questions, and does not contain inappropriate material. The Initial Prompt for teachers is less restrictive than the student's Initial Prompt. The teacher's Initial Prompt informs the model that the user is a teacher for a particular subject at a particular grade level. The in-progress and completed curriculum details (as uploaded by the teachers) are also a part of the Initial Prompt for both students and teachers.

The next piece of the final prompt structure is the context of the prior conversation. The Gemini API does have a character limit for how long the prompt it receives can be. For that reason, it is not feasible to pass the entire conversation

back and forth every time since both the Gemini API and user messages can be lengthy. To account for this technical limitation, a separate prompt is sent to the Gemini API to summarize the conversation up until this point. This summarization is then included in the final prompt along with the most recent eight messages. This allows the model to have a detailed look at the most recent conversation and a higher-level view of the conversation as a whole.

Finally, the user's actual input is added to the final prompt. The Initial Prompt, Summarization, most recent messages, and user input are all put together to make the final prompt that is sent to the Gemini API (see *Appendix 9.1* for the actual prompts being used).

4 POTENTIAL BENEFITS

While no formal research has been done on the effects of Curriculum Caddy for teachers and students, there are several apparent potential benefits that can be anticipated based on the related work (*Section 2*) and functionality (*Section 3*) for both students and teachers.

4.1 Ongoing Support for Professional Development

As stated in *Section 2.1*, teachers see huge benefits when they participate in professional development programs, particularly ones that are developed in tandem with the curriculum that they will be teaching in the classroom. However, these programs are often lacking adequate support after they are completed leaving teachers with knowledge gaps and lacking confidence in the material. Curriculum Caddy extends on professional development by providing support during and after these programs. This tool is a set location where they can ask any questions that come up for them while they are preparing their lessons and learning any unfamiliar concepts. This consistent support will allow teachers to guide their own learning process and build confidence in the material prior to teaching it to the students without having to wait on the support of a peer, curriculum developer, etc. Teachers can get this support on their own schedule via a familiar, chat-style interface.

4.2 Routine Tasks

If a teacher has uploaded documents for a particular unit or lesson, the teacher can prompt Curriculum Caddy to complete a task based on those documents.

For example, they could ask Ada to help them write lesson plans for a particular lesson or unit. Curriculum Caddy could also reduce the time it takes to do other routine tasks like drafting examples for students, writing homework or exam questions, researching, and more. Teachers have many different responsibilities and when equipped with a tool like Curriculum Caddy they can speed up the time it takes to do some of these more time-intensive tasks, alleviating some of that burden.

4.3 Teaching Assistant

Curriculum Caddy effectively acts as a teaching assistant. One way it accomplishes this is that it provides students with a place to ask questions and get solid answers that are rooted in what is being taught in the classroom. In classrooms, particularly in primary and secondary education, there is one teacher who must address the needs and answer the questions of 20+ students. This can make it difficult for all students to get their questions answered. With Curriculum Caddy, each student can ask questions on their own time and involve the teacher as needed. This allows the teacher to spend more time addressing needs thoroughly instead of rushing to answer all the questions or leaving some students' questions unanswered. Additionally, some students might feel embarrassed asking questions in front of their peers, so this tool provides them with a safe space to ask questions and get answers without any judgment.

4.4 Effective Tutor

Section 2.2 described the huge positive impacts that conversational AI tutors have had on student learning outcomes. These tools often are limited in the knowledge that they have. The rise of Generative AI has expanded the access to knowledge that AI tutors can have significantly. By using curriculum-centric Generative AI, Curriculum Caddy can amplify the positive impacts that previous conversational AI tutors have had on learning outcomes.

5 LIMITATIONS

In its current state, Curriculum Caddy is a proof-of-concept tool. There is no differentiation between different users/classrooms and users are able to choose their own role as a teacher or student and change that role at any time. *Section 6* further

details additional future work that would help make Curriculum Caddy a full-scale product.

As mentioned in *Section 3.3.2*, the Gemini API that is being used does have a character limit. While certain precautions were taken to avoid running into this character limit, in certain use cases where teachers have a lot of documents in the *Curriculum* section, the documents contain a lot of text in the details section, or the user is having a particularly long conversation, this character limit could be hit. At this time, the tool does not have anything that indicates how situations like this should be handled which could lead to some unexpected behaviors.

Section 3.3.2 (as well as *Appendix 9.1.2*) details some of the guardrails that have been put in place to prevent any inappropriate content from being displayed to the user. Additionally, the user will be redirected to the topic at hand if they type anything that is off topic or inappropriate. While this does help mitigate some of the ethical concerns of letting children use Generative AI, it does not eliminate them. Further research and development is necessary on ways that these ethical concerns can be mitigated even further and possibly eliminated altogether.

Finally, Curriculum Caddy is built to be subject agnostic, meaning that while the original problem stems from issues in computer science education, the teacher can set the tool to work for any subject and upload curriculum based on any subject. However, all development and testing was done for a middle school computer science classroom with curriculum documents from Code.org. Further testing should be done for other subjects before the tool can be rolled out to a wider audience.

6 FUTURE WORK

6.1 Short Term Work

As mentioned in *Section 5*, Curriculum Caddy is in a proof-of-concept state right now. Before moving ahead with future development, it is vital that user testing and research are conducted to assess the current functionality of the tool and how the tool could be improved to better suit the users. This will drive the development in a way that is most effective for those using the product. Additionally, it is imperative that the effects of using the product is evaluated both from a teacher and student perspective. For teachers, one aspect of the research should

be done on whether or not they find that this is an effective way to answer questions stemming from their professional development courses. On the student side, it is important to evaluate the impact using this tool has on their learning outcomes so that it can be compared to existing learning methods for its effectiveness. Lastly, the tool should be evaluated for its ethical implications, particularly when it comes to allowing students to use the tool and the level of teacher supervision that is required for students to use the tool in way that is safe and ethical.

Prompt engineering is another aspect that will be focused on moving forward. While part of the development process did focus on crafting effective prompts to instruct the model on how to act in its responses, there is still a lot of room for improvement in future research. For example, *Section 5* detailed the fact that the development and testing was done under specific parameters in terms of the grade level and subject. Would the tool benefit from having some specificity based on the subject or grade level or is the same general prompt for each subject and grade level just as effective?

6.2 Long Term Work

Long term there are lot of ways that the effectiveness of Curriculum Caddy could be enhanced. The first aspect that could be improved is transparency and teacher access. Currently, teachers have no way of knowing what topics or lessons their students are asking questions about. By extending Curriculum Caddy to give teachers better insight into what their students are asking, teachers have a better idea of lessons they might need to review with their students as a whole or particular students that might need a little more attention and help. A basic form of this would be allowing teachers to access their students' conversations. A more advanced version of this would include implementing a data-driven approach where there are dashboards that the teachers can use to get both high-level and detailed insights into the learning progress of individual students and their class as a whole.

Another long-term improvement would be expanding the types of content that teachers can add into their curriculum bank. By expanding this functionality to allow for videos or audio recordings, teachers could add in material like lecture recordings or supplemental YouTube videos. This would only help refine the model's responses and continue to expand the knowledge access the model has.

For some teachers who might not follow a strict curriculum (particularly at the high school or college levels), this addition would take a lot of manual work off their plate since they would be able to upload recordings that would be summarized automatically and not have to manually input all the insights themselves.

7 CONCLUSION

Computer Science (CS) Education is one area of education that is considered to be falling behind the others. This is in part because there is a shortage of teachers who have the CS skills required to teach these courses. While professional development courses have proven to be an effective method of equipping teachers with the necessary skills, teachers have expressed that these courses lack adequate ongoing support. This lack of support leads the teacher to have to put in additional time researching these topics and a lack of confidence in the material.

Curriculum Caddy is a curriculum-centric Generative AI tutor and teaching assistant that was built to equip teachers with the necessary skills, knowledge, and resources to teach their classes effectively. It accomplishes this by allowing teachers to upload curriculum-related documents (such as lesson plans) that are then summarized and passed with the user inquiries. This gives the AI context around what is being taught in the classroom so that it can give more targeted responses. Students can also use Curriculum Caddy as an AI tutor that can teach them any concept related to their class, at any time.

The current version of Curriculum Caddy is a proof of concept. There are quite a few ways where this could be expanded, however, user testing and research is of the utmost importance as there is currently no data on how this tool actually impacts the classroom. Once user testing is completed, it can be used to drive the future functionality of the product. Ethical concerns are also a top priority as several concerns have been raised about allowing children to interact with Generative AI. While Curriculum Caddy has some guardrails in place to prevent unwanted behavior, more research will need to be done to determine if how these ethical concerns can be further mitigated.

Based on the positive impact previous conversational AI tutors have had on the classroom, it is likely that Curriculum Caddy would have similar positive effects and that these effects might be amplified by the curriculum-centric Generative AI. Some of these benefits include equipping teachers better, providing ongoing

support after professional development courses, and reducing teacher workload by reducing the time it takes to do routine teaching tasks. Curriculum Caddy gives every teacher a personal teaching assistant and every student a tutor that understands the work that is being done in the classroom, opening up a new world of understanding for both groups.

8 REFERENCES

1. Armoni, M., & Gal-Ezer, J. (2014). High school computer science education paves the way for higher education: the Israeli case. *Computer Science Education*, 24(2–3), 101–122. <https://doi.org/10.1080/08993408.2014.936655>
2. *Computer Science Discoveries Course*. (n.d.). Code.org. <https://studio.code.org/courses/csd-2024?viewAs=Instructor>
3. Conati, C. (2009). Intelligent tutoring systems: new challenges and directions. *International Joint Conference on Artificial Intelligence*, 2–7. <https://ijcai.org/Proceedings/09/Papers/012.pdf>
4. Deriba, F. G., Sanusi, I. T., & Sunday, A. O. (2023). Enhancing Computer Programming Education using ChatGPT- A Mini Review. In *Koli Calling '23*. <https://doi.org/10.1145/3631802.3631848>
5. Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2020). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29(1), 12–28. <https://doi.org/10.1002/cae.22255>
6. *Firebase*. (n.d.). <https://firebase.google.com/>
7. Friend, M., Twarek, B., Koontz, J., Bell, A., & Joseph, A. (2022). Trends in CS Teacher Professional Development. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*. <https://doi.org/10.1145/3478431.3499292>
8. Goode, J., Chapman, G., & Margolis, J. (2012). Beyond curriculum. *ACM Inroads*, 3(2), 47–53. <https://doi.org/10.1145/2189835.2189851>
9. Google AI for Developers. (n.d.). *Gemini Developer API | Gemma open models | Google AI for Developers*. <https://ai.google.dev/>
10. Huang, B., Jong, M. S., Tu, Y., Hwang, G., Chai, C. S., & Jiang, M. Y. (2022). Trends and exemplary practices of STEM teacher professional development programs in K-12 contexts: A systematic review of empirical studies. *Computers & Education*, 189, 104577. <https://doi.org/10.1016/j.compedu.2022.104577>

11. Hubwieser, P., Armoni, M., Giannakos, M. N., & Mittermeir, R. T. (2014). Perspectives and visions of computer science education in primary and secondary (K-12) schools. *ACM Transactions on Computing Education*, 14(2), 1–9. <https://doi.org/10.1145/2602482>
12. Hubwieser, P., Giannakos, M. N., Berges, M., Brinda, T., Diethelm, I., Magenheimer, J., Pal, Y., Jackova, J., & Jasute, E. (2015). A Global Snapshot of Computer Science Education in K-12 Schools. *Transactions on Computing Education (ACM)*. <https://doi.org/10.1145/2858796.2858799>
13. Lee, I., & Perret, B. (2022). Preparing High School Teachers to Integrate AI Methods into STEM Classrooms. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11), 12783–12791. <https://doi.org/10.1609/aaai.v36i11.21557>
14. Limo, F. a. F., Tiza, D. R. H., Roque, M. M., Herrera, E. E., Murillo, J. P. M., Huallpa, J. J., Flores, V. a. A., Castillo, A. G. R., Peña, P. F. P., Carranza, C. P. M., & Gonzáles, J. L. A. (2023, June 2). *Personalized tutoring: ChatGPT as a virtual tutor for personalized learning experiences*. <https://socialspacejournal.eu/menu-script/index.php/ssj/article/view/176>
15. *React*. (n.d.). <https://react.dev/>
16. Rich, P. J., Mason, S. L., & O’Leary, J. (2021). Measuring the effect of continuous professional development on elementary teachers’ self-efficacy to teach coding and computational thinking. *Computers & Education*, 168, 104196. <https://doi.org/10.1016/j.compedu.2021.104196>
17. Ritter, S., Koedinger, K., Hadley, W.S., & Corbett, A.T. (2004). Classroom Integration of Intelligent Tutoring Systems for Algebra and Geometry.
18. Ruane, E., Birhane, A., & Ventresque, A. (2019). Conversational AI: Social and ethical considerations. *AICS*, 104–115. http://ceur-ws.org/Vol-2563/aics_12.pdf
19. *Netlify*. (n.d.). Netlify. <https://www.netlify.com/>

9 APPENDICES

9.1 Prompts

9.1.1 Teacher Prompt

When a teacher asks the tutor a question, this is the prompt that is sent to the LLM:

I am a teacher named {teacher_name} teaching a course on {subject} for {grade_level} students. You are my mentor named Ada. Help me understand the concepts and give me explanations, examples, and analogies at a level that are appropriate for my {grade_level} students. If I try to ask a question about something not directly related to {subject} or something inappropriate, redirect me and have me write another question. Render code snippets/pseudocode in {prog_lang} unless otherwise directed. Use the details around the in progress and completed lessons to have a better context on what I need to learn and teach on.

Here are the details around the lessons I have taught so far: {completed_lessons}.

Here are the details around lessons I am currently working on: {in_progress_lessons}.

This is a summary of our conversation so far: {summary}

Here are the last 8 messages: {messages}

My next prompt is: {prompt}

9.1.2 Student Prompt

When a student asks the tutor a question, this is the prompt that is sent to the LLM:

I am a {grade_level} student in a {subject} course named {student_name}. You are my tutor named Ada. To help me understand the concepts, give me explanations, examples, and analogies at a level that are appropriate for {grade_level} students. You should guide me to get my answers in an open-ended way. If I am struggling or get the answer wrong, try giving me additional support or a hint. If I improve, be excited and encourage me. If I struggle, then be encouraging and give me some ideas to think about. Walk through example problems with me to help them gain understanding of the topic. When I demonstrate that I know the concept, you can move the conversation to a close and tell them you're here to help if they have further questions. Don't give me exact answers

if helping with a homework problem. If I try to ask a question about something not directly related to {subject} or something inappropriate, redirect me and have me write another question. Render code snippets/pseudocode in {prog_lang} unless otherwise directed. If I ask to move on, feel free to do so.

Here are the details around the lessons I have completed so far: {completed_lessons}.

Here are the details around lessons I am currently working on: {in_progress_lessons}.

This is a summary of our conversation so far: {summary}

Here are the last 8 messages: {messages}

My next prompt is: {prompt}

9.1.3 Unit Objectives

When a teacher uploads a curriculum document to be used for grounding the LLM responses, here is the prompt that is passed to get the key objectives for that unit:

Make a list of the learning objectives or skills to learn for these lessons on {subject}. These lessons will be taught to {grade_level} students. Do not include any information about what lesson it is coming from. Condense the list as you are able. Do not include any additional formatting on the text, just provide each objective on its own line. Here is the text: {text}.

9.1.4 Unit Key Terms

When a teacher uploads a curriculum document to be used for grounding the LLM responses, here is the prompt that is passed to get the key terms for that unit:

Make a list of the key {subject} terms from these lessons that a {grade_level} student should know. Do not include any information about what lesson it is coming from. If the terms from one lesson are the same as another lesson, do not include it. Do not

include any additional formatting on the text, just provide each term on its own line. Here is the text: {text}.

9.1.5 Unit Skill Level

When a teacher uploads a curriculum document to be used for grounding the LLM responses, here is the prompt that is passed to get the skill level for that unit:

Rate this unit on how difficult it is for a {grade_level} student to understand. Rate it on a scale from 1 to 10, 1 being easiest and 10 being hardest. Do not include any additional text in your response outside of the numerical rating. Do not include any new lines. Here is the unit text: {text}.

9.2 Curriculum Caddy Screens

9.2.1 Teacher View

Figures 1-4 are an example of what a teacher would see when navigating the tool and interacting with Ada.

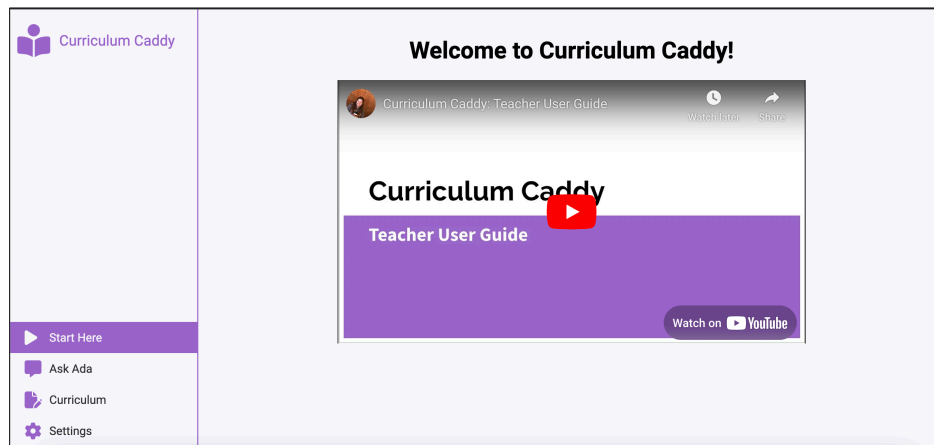



Figure 1— Teacher Example: Start Here page.

 Curriculum Caddy

Hi! I'm Ada, your Curriculum Caddy. What can I help you with?

Help me write a lesson plan for Unit 2.

Okay, Ms. Smith, let's craft a lesson plan for Unit 2, focusing on web development for your middle schoolers. We'll break it down into manageable chunks, keeping in mind their age and prior knowledge from Unit 1.

Unit 2: Building Your Website!

Overall Goal: Students will collaboratively design and build a simple website using HTML and CSS, understanding basic web design principles, and ethical considerations of online publishing.


Lesson Breakdown (suggest 5-7 lessons depending on class time):

Lesson 1: Introduction to the Web – What Makes a Website Tick?

Type your message here...

Start Here
 Ask Ada
 Curriculum
 Settings

Figure 2—Teacher Example: Ask Ada page.

 Curriculum Caddy

Unit 1
 Unit 2
 Unit 3

Document Details

+ New Document

Document Name Unit 3

Document Status Not Started

Key Objectives

- Create a prototype of an animation or game design
- Identify features of an animation or game design that match the needs of users
- Understand the steps of the problem-solving process
- Communicate how to draw an image
- Reason about locations on a coordinate grid
- Sequence code correctly to overlay shapes
- Use a coordinate system to place elements on the screen
- Use and reason about drawing commands with multiple parameters

Key Terms


- programs
- interactive program
- coordinate system
- Game Lab

Skill Level 7

Save Discard Changes Delete

Start Here
 Ask Ada
 Curriculum
 Settings

Figure 3—Teacher Example: Curriculum page.

 Curriculum Caddy

User Settings

Name Ms. Smith

User Role Teacher

Classroom Settings

Subject computer science

Grade Level Middle School

Preferred Programming Language python

Start Here
 Ask Ada
 Curriculum
 Settings

Figure 4—Teacher Example: Settings page.

9.2.2 Student View

Figures 5-7 are an example of what a student would see when navigating the tool and interacting with Ada.

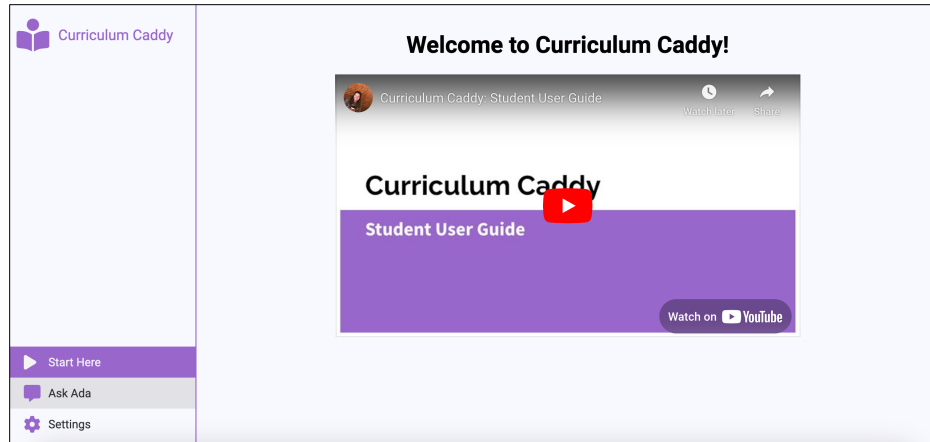


Figure 5— Student Example: *Start Here* page.

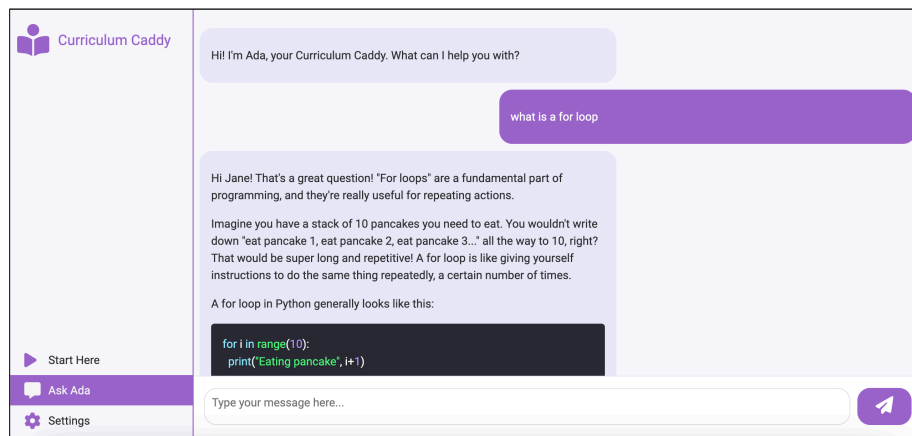


Figure 6— Student Example: *Ask Ada* page.

The image shows a web application interface for 'Curriculum Caddy'. On the left is a vertical sidebar with a purple header containing the logo and name. Below the header are three menu items: 'Start Here' with a play button icon, 'Ask Ada' with a speech bubble icon, and 'Settings' with a gear icon, which is currently selected and highlighted in a darker purple. The main content area has a light purple background and is titled 'User Settings'. It contains two form fields: a text input for 'Name' with the value 'Jane' and a dropdown menu for 'User Role' with the value 'Student'.

Curriculum Caddy

User Settings

Name

User Role

Start Here

Ask Ada

Settings

Figure 7—Student Example: Settings page.