

1. Statement of Purpose

Learning and becoming an expert in a new programming language or skill is a long-term, often challenging process for those with limited or no experience. Although daily efforts are doubtless essential for success in a month-long or even year-long learning program, learners often find it hard to maintain consistent motivation and progress. Sometimes people just give up. Sticklers are rare. Platforms like Udemy, Coursera and Edx are wonderful for sticklers, but for those of us who give up when we get bored, we need a Stickler to sit on top of our head and get things done. Apps till date have been all about getting people hooked up, with the recent trend of those minimalism apps that help you reduce screen time and stuff, but we digress! The point is, we need a Stickler of an app!

So how do we do that? How do we help somebody to just “stick to it”, become indefatigable? We propose the web application *Streak* (for now, which we will eventually rename). This app offers a two-fold solution we call *motivation-driven active learning* and *passive reminders*. For the motivation-driven active learning, we will create a skill-tree-style progress visualization interface to display the learning roadmap - the greater your daily streak , the deeper the roots of the tree. The greater the number of leaves, the more proficient you are, with each leaf representing what the user learns. As learners progress, they have the ability to add new skills to this tree, and even build new trees, motivating them to explore subtasks and sub-skills based on what they've learned. This lets users clearly see their achievements, motivating daily efforts toward their goals. For passive reminders, we plan to notify users if they miss a day of learning. These reminders will include brief motivational messages and visual cues to help maintain focus and motivation. So, from the makers of [cse210-tinyfoot-11](#), (which we are not proud of (at all), but because we wanted to sound theatrical) presenting: *Streak*...

2. User Profiles

Angry Atishay (He/Him/His)



Age: 22

Education Background: Kickass tester - destroys software developers deployment in

testing sessions. Did his bachelors from the Avengers Institute of Technology, Reed Richards School of Engineering. Trying to make a name in the field of software development to make better software that testers will enjoy testing. Ah, youth - how their blood boils!

Job: No sir! He's a grad student at a prestigious university. He is still applying for internships, do you have any for him?

Hobbies: Binge watches anime and explores new places. He just moved to an area with

tons of beaches so he is learning how to swim.

His Problems: Atishay wakes up every day after 6 hours of sleep (doctors recommended 7) to attend classes, which are way too advanced for him. He spends his entire break and after-class hours trying to grasp new concepts and keeping up with assignments. However, due to these priorities, he doesn't have enough time to practice the fundamentals of programming, which leaves him lacking the confidence needed to tackle problems quickly. He keeps ordering takeout and doesn't cook his own food. As a result, despite understanding the concepts, he's only able to complete half his assignments, impacting his grades. The kid really needs to step up and discipline himself.

Teacher Tanner (He/Him/His)



Age: 32

Educational background: BS in Computer Science from University of Merryland, grew up in the midwest on his parent's farm.

Job: Tanner works for a startup doing backend development, he also works for a company that provides tutors to kids who are learning how to code. He has to write his own lesson plans for the kids. Every coding bro ever, right?

Hobbies: Tanner loves to DJ and mix, and spends lots of time looking for songs on Soundcloud and booking gigs to play at. He's also a weightlifter, and works out on weekends. That explains the beef up.

His Problems: Tanner wakes up early to log into his job at ABCDEFGHIJK Systems Inc., a hot new tech startup that requires backend developers. Since he works remotely, he also waits in a queue of tutors to help students on the side. He's got a DJ gig booked for next week, but also a big project due for the startup. He works on the code for ABCDEFGHIJK Systems Inc. every day for hours, and since he's also having to spend time helping students learn to code, he realizes at the end of the week that he's very underprepared for the DJ set! Oy vey! He performs terribly because he didn't practice and gets banned from the nightclub.

HR Hamlin (Prefer not to disclose)



Age (Classified): 42 (Nobody knows their real age)

Education Background (Classified): Master's degree in Human Resources Management, attended the prestigious Corbell University. At least their LinkedIn profile says so.

Job (Classified): Senior HR Manager at a FoogleSoft Inc - hmm, sounds legit.

Hobbies (Not classified): Avid golfer, passionate about classical music (especially Beethoven), and an enthusiastic collector of vintage fountain pens. Known among friends for hosting elaborate wine-tasting parties. Straight out of the game of Clue.

Personal Details (Classified): Come from an upper-middle-class family but have climbed the corporate ladder on their own merits - rumor has it that they're fighting crime in Hell's Kitchen at night alongside Daredevil. Known for their meticulous dress code, they maintain a collection of tailored suits and are rarely seen without their gold-rimmed glasses. They drive a Mercedes-Benz to work but have a personal assistant handling scheduling and day-to-day errands. Took up coding because they wanted to conquer the "hot" world of AI.

Their Problems: Hamlin are currently studying for an additional degree in CS (as if their qualifications aren't enough), and encounter unexpectedly heavy workload. They decide to leave weekends open for studying and coding, so they can still work their day job as HR, alongside fighting crime at night. After a few weeks, they realize that the weekends are not enough time to complete assignments alongside crime fighting and golfing. Grad school hits differently - again.

Accountant Angela (She/Her/Hers)



Age: 35

Education background: MBA from the New York University

Jobs: Angela works for an accounting firm in NP city, playing with crazy huge figures everyday.

Hobbies: Cooking, she has a YouTube channel teaching people how to cook. Has 2 kids aged 8 and 10. Angela manages her career and life pretty well - her husband's pretty happy with her. She is learning to code for her cooking website - and she is not exactly acing it, she lacks motivation, like Atishay.

Her Problems: She has to work late due to an issue with the accounts, several high-profile clients needed additional attention. She also has an upload schedule for cooking videos, but missed the upload and people in the comments are wondering where she is. She feels dejected that amidst all this, she has no time to work on her website.

Broker Bill (He/Him/His)



Age: 35

Education background: Bachelor's degree in accounting, from a rich family

Jobs: Trader @ Silverman Sazs, bonds broker, trades \$5B bonds everyday. JB Organ is trying to poach him for the past few years.

Hobbies: Plays League of Legends, hates Faker. He's trying to write cheat codes, and has clients depending on him to provide cheats for LoL. He has a personal limo and a driver who drives him around (he's richie rich rich). Broker Bill works long hours but has to squeeze in time to develop cheats for league of legends.

His Problems: Riot games releases a patch that makes most of his cheats break, and he has lots of angry customers wanting upgraded cheats. He plans out a schedule for fixing the cheat software, but when his proposed due date arrives, he finds that the software is nowhere near done and his clients are madder than mad hettie.

3. Risk Analysis & Rabbit Holes

- a. *Visualizing and Aestheticizing the Tree Structure:* When tasks and subtasks accumulate, the tree can become overly complex, leading to cluttered interfaces and a poor user experience.

Risks:

- As the number of tasks grows, the tree nodes and levels increase, possibly causing screen overflow, making it hard to navigate and even affecting load times.
- Complex tree structures can be difficult to display cleanly and understandably within a single view.

- b. *Marking Task Completion Status:* Marking task statuses (completed, in-progress, etc.) might lead to a visually overwhelming interface, especially with large trees.

Risks:

- Representing completion in a way that's clear and doesn't overcrowd the tree.
- Relying heavily on icons or colors can confuse users or detract from readability if not used consistently.

- c. *Dynamic Tree Modification and Consistency Management:* Allowing users to dynamically add, delete, or modify nodes can introduce challenges to maintaining a stable and consistent tree structure.

Risks:

- Ensuring that modifications like adding, removing, or moving nodes don't destabilize the structure.
- Keeping the tree structure updated across multiple users or devices in real time.

- d. *Data Storage and Loading Performance:* As the tree structure grows, data loading and query performance could slow down, affecting user experience.

Risks:

- Efficiently storing and querying large tree structures, especially for deeply nested tasks, without causing excessive query times or memory usage.
- Optimizing tree rendering to avoid frequent and resource-intensive re-renders.

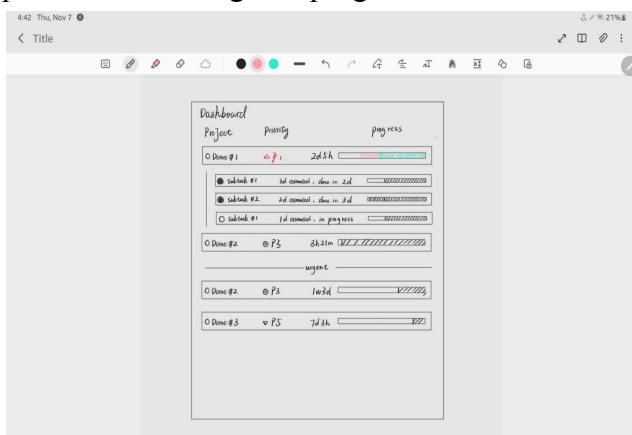
e. *UI Compatibility and Adaptability*: Displaying complex trees on different devices (e.g., mobile, tablet, desktop) can lead to inconsistency and usability issues, particularly on smaller screens.

Risks:

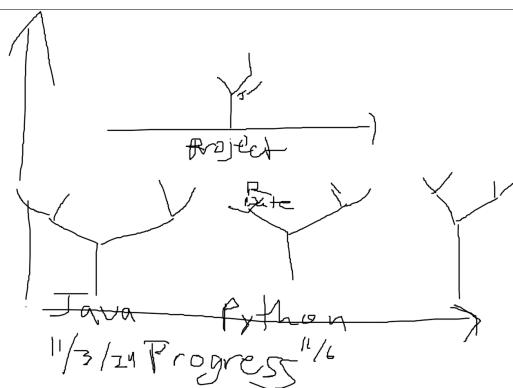
- Adapting to various screen sizes to ensure the tree structure displays effectively across devices.
- Simplifying tree navigation on small screens to minimize the need for excessive scrolling or zooming.

4. Visual Representations

Here's an early iteration of our vision (refer Iteration 1). This solves the issue of time management, but has some holes. Also, how would we support longer goals? Maybe the overall process of learning as a programmer?

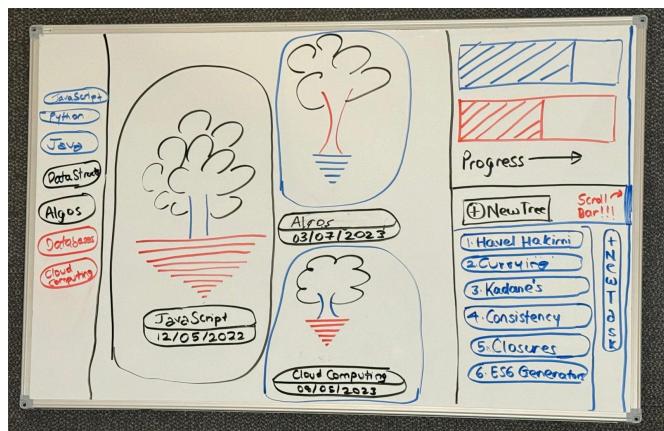


Iteration 1



Iteration 2

Our next iteration - Iteration 2 - A “hall of fame” of your achievements as you learn or work on projects. Everything is preserved in the timeline, and starting a new tree or building is quick and easy. As your trees grow, topics you are learning are listed and preserved, along with the dates. If you do a little bit of learning every day, your motivation will increase.



Iteration 3 - A real fat-marker sketch

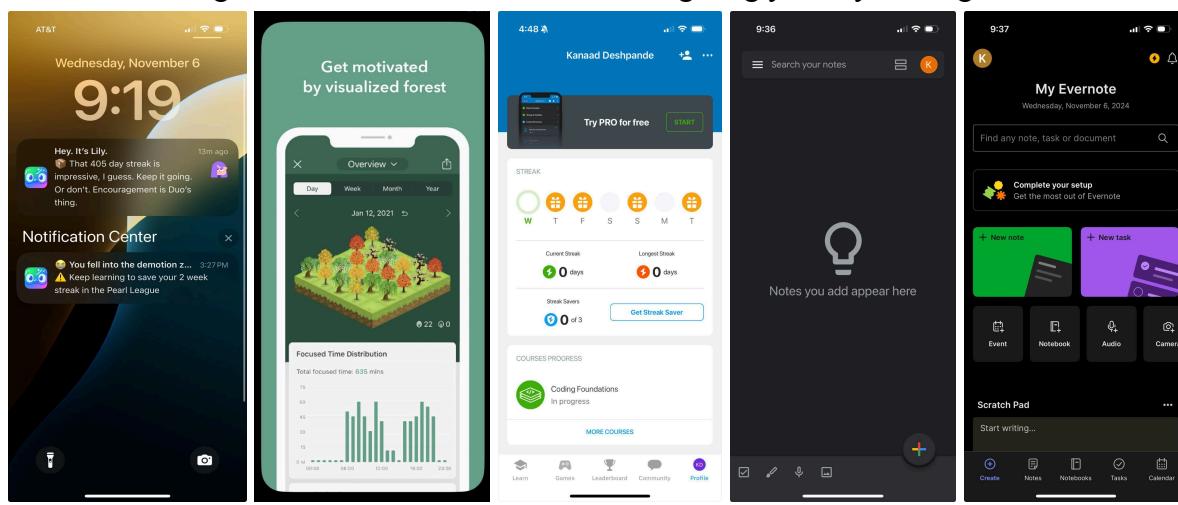
In this iteration, we have a navbar that has the skills that the user is trying to learn. Each tree is in a tile - taking inspiration from the tiles that we see on Pinterest. The tree shows the start date and the bigger the root - the bigger your streak, the more the leaves, the more areas you have covered. The progress is also tracked on a progress bar. You can create a new learning tree or pick up a task from the ones that already exist or create a new one. Note that we will be having multiple scroll areas - one for the tree tiles, one for the task list, and one for the progress bar - inspiration? Spyder - from Anaconda.

5. Market Research

Our market research consisted of a study of applications that we have previously used to integrate elements that we liked into our product. We started with the adorable green owl that everybody loves. The app that makes you keep your Streak no matter what. Yep, Duolingo. That's how a team member of ours got to the 500 day streak!



<https://www.duolingo.com> - an app that utilizes a streak/growth system. While this app is specific to language learning, it is based on a principle that can be used to motivate programmers to continue learning with the constant notifications badgering you day and night!



Duo Notifs

Forest

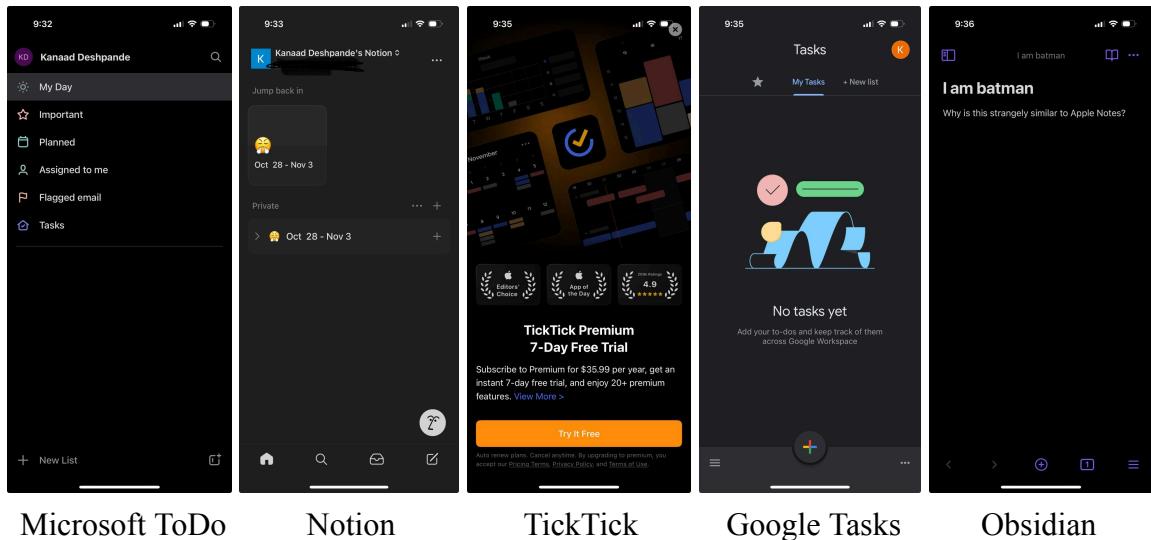
Sololearn

Google Notes

Evernote

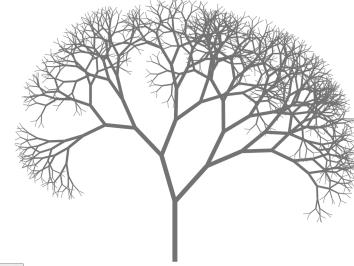
<https://www.forestapp.cc> - An app that focuses on productivity motivation, by watching a small tree grow. We want to capture the motivational aspect of forest in our app. This app inspired the trees in our product.

<https://www.sololearn.com/en/>: The solelearn app is wonderful - it does exactly what duo does. Only problem? It's too restrictive, you gotta learn stuff their way, you have no freedom to navigate the intricacies of the world of programming. Also it's paid and really costly, Udemy works better.



Way too many apps, drove us crazy!

So the gist of it is that, there were lots and lots of apps out there that do the “productivity” thingy. They take notes for us, they have voice activated commands, and you can do pretty much anything that you can dream of with these features - but they’re only useful if you want to be productive. If you want somebody to be productive for you (although there’s a limit to it!), you need a minimalistic reminder and progress tracker, and that’s the job of *Streak*.



<https://app.peterrcook.com/d3-tree/> - This algorithm creates a tree similar to what we envision. Possible tool to use for our trees. We have included this in market research as a means to develop our product. Same for: <https://www.florasynth.com/>.

5. Conclusion

As we sat, wondering what we could do with our limited web development knowledge, the ideas vague, the process unclear, we realized that the target audience for our product were supposed to be in the same boat that some of us were. Some of us were transitioning from non-CS backgrounds to CS, so while the initial journey was a wee bit difficult, we realized that we could leverage our lack of knowledge and experience to our advantage - and that's what we did. The Streak app, (will be renamed eventually) helps build a streak, and helps you become productive. We won't encroach your freedom, go learn whatever you want, we won't overwhelm you with mind boggling voice command features or high tech stuff, we'll just sit on your head and ensure you get things done. Easy peasy. Therein, we conclude our pitch. Team Hackstreet Boys, signing off!