Final Report: Engaging Science Education for Teens

> by Wandile Ndaba (31 May 21)

1. Introduction

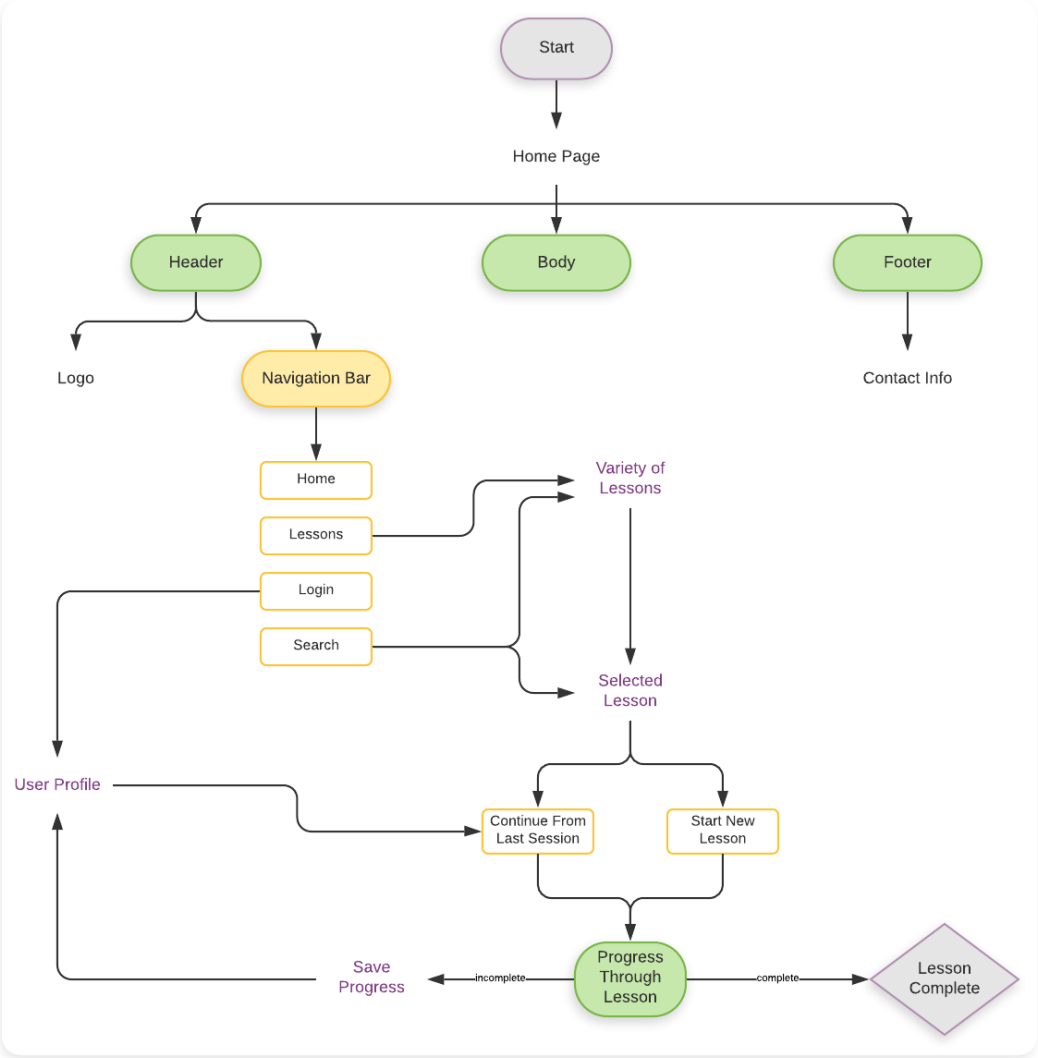
- *Goal:* The goal of this project was to create a **website** that teaches **STEM** *(Science, Technology, Engineering, Mathematics)* material to teenage students.
- It was intended to be narrative-driven it's writing tone, highly visual and applying best practices and science-informed teaching strategies. The plan was to implement an audio voice-over that students can activate whenever they get stuck on a topic.
- *The results:* The features intended for inclusion to aid student's learning, included: 1. an audio player on the page at all time; 2. a progress bar; 3. a responsive design;

2. Design and Implementation

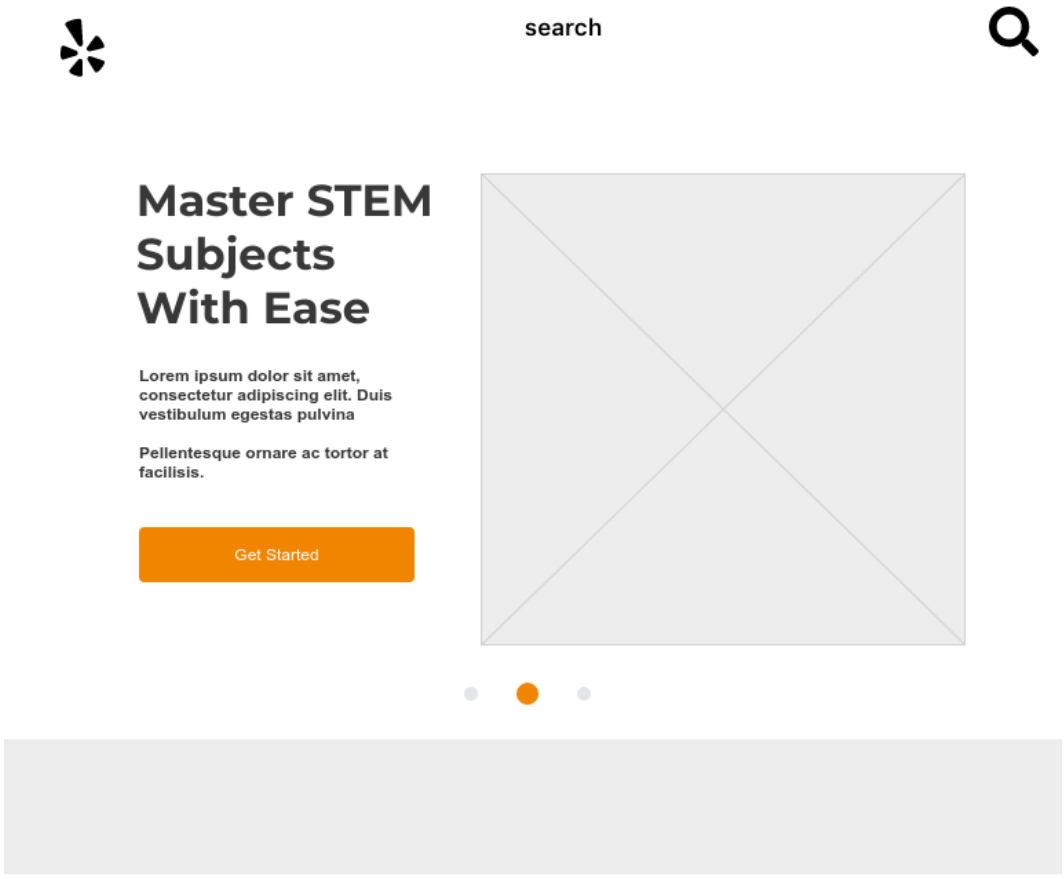
- The design for the website, inspired by other existing learning applications (eg. Coursera), was done in Adobe illustrator and finally wireframed using proto.io.
- To help flesh out the visual idea and communicate it better to others, I drafted/prototyped two pages from the website - the homepage and a single lesson page.

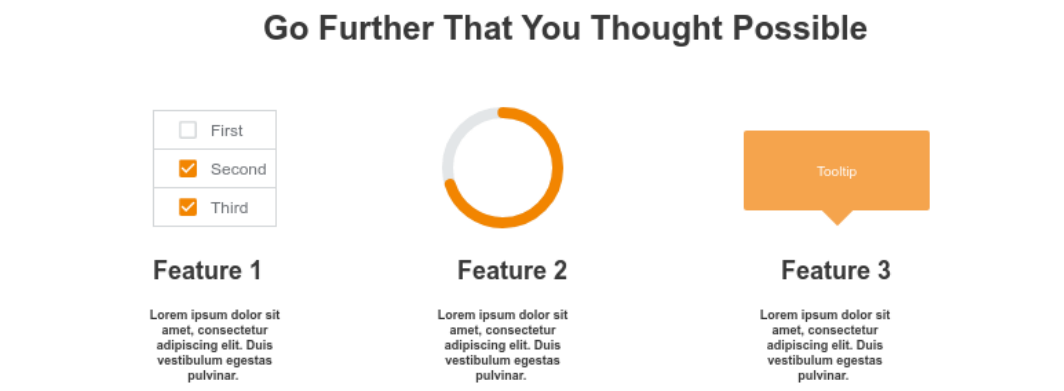
2.1.0 Typical User Flow

User experience journey for completing a lesson

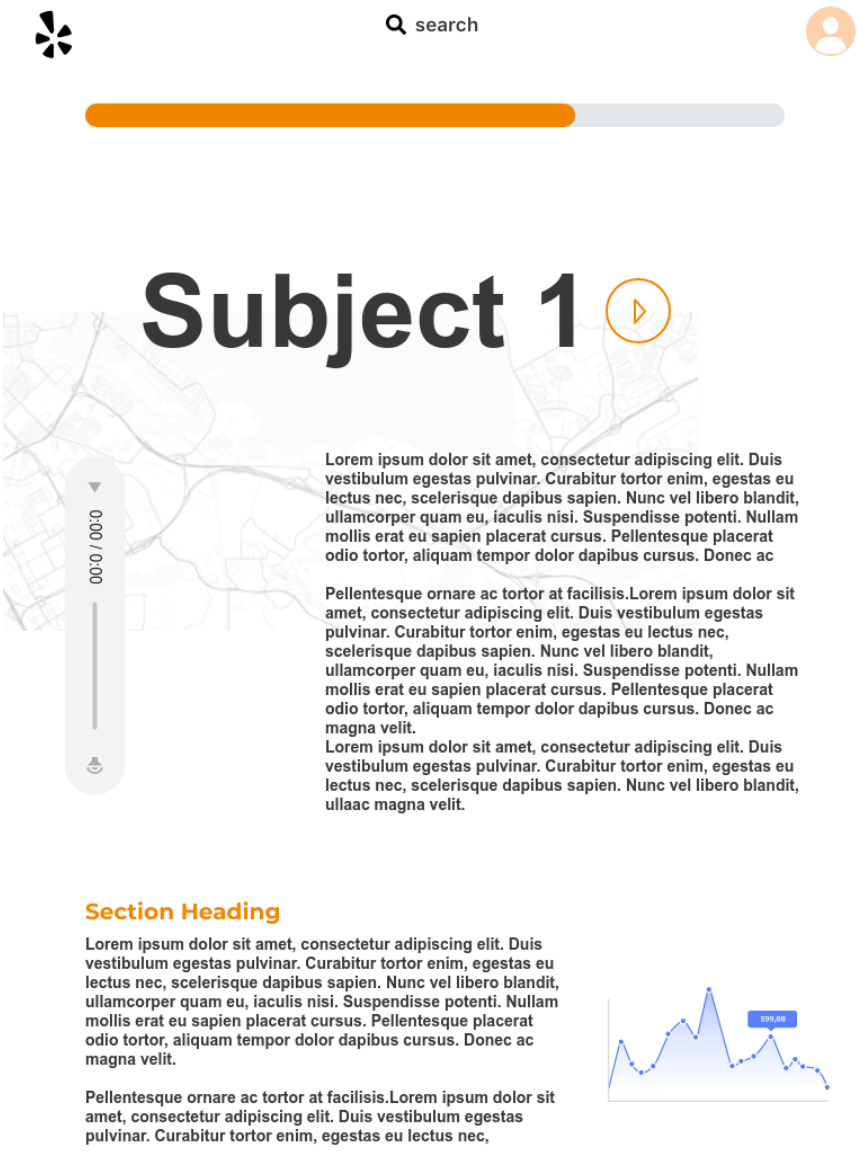


2.1.1. Homepage Mockup





2.1.2. Single Lesson Page Mockup

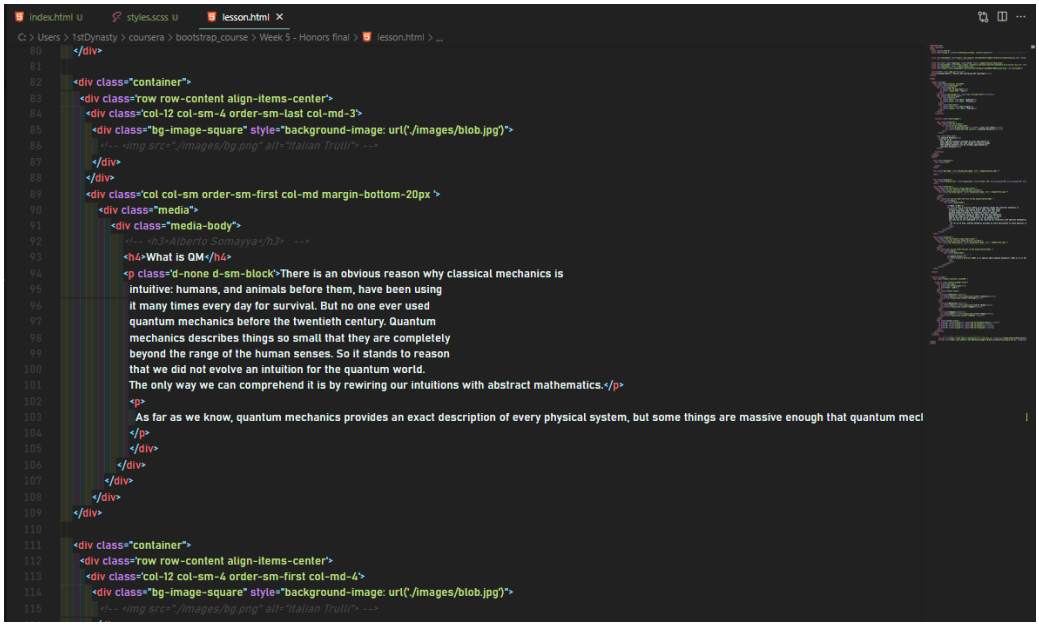


- After research phase was complete, I was left with a compelling design. but it included many elements/moving pieces and features, most of which I had never implement in code before. This meant a potentially for confusion, which would lead predictably to difficulties with the project) So I broke down the final design, **listing** out all the necessary **components**.
- Over the week I gradually found **resources**, such as online tutorials, that would help realise the functionality of the website.
- Production: Week 1 -The content.
    - First, I picked a subject for the preview lesson page. This had to be an exemplar for the type of course material students should expect

and the difficulty level. study the subject enough to distill it. Create any illustrations needed for the

- I also created a bibliography using Mendeley, of for resources that I would use for the content of the lesson page.

- production: Week 2 - Coding
  - I coded the website using VS Code alongside a few of my favourite extensions.



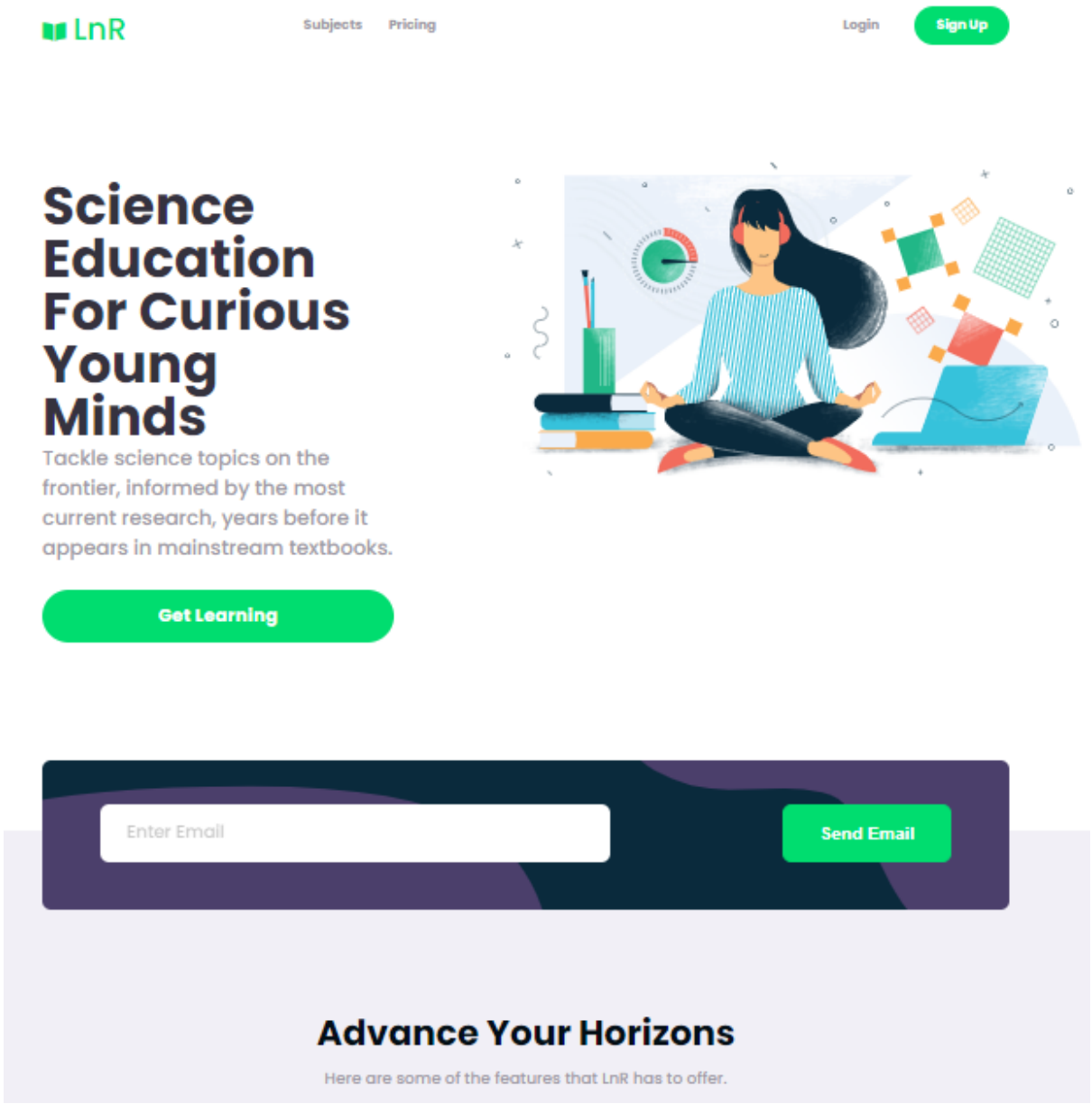
- There were no libraries or modules utilised in this project, everything was coded from the ground up. My reasoning was that, given this was a learning opportunity more that anything else, the goal was solve as many of the challenges myself.

- post-production:

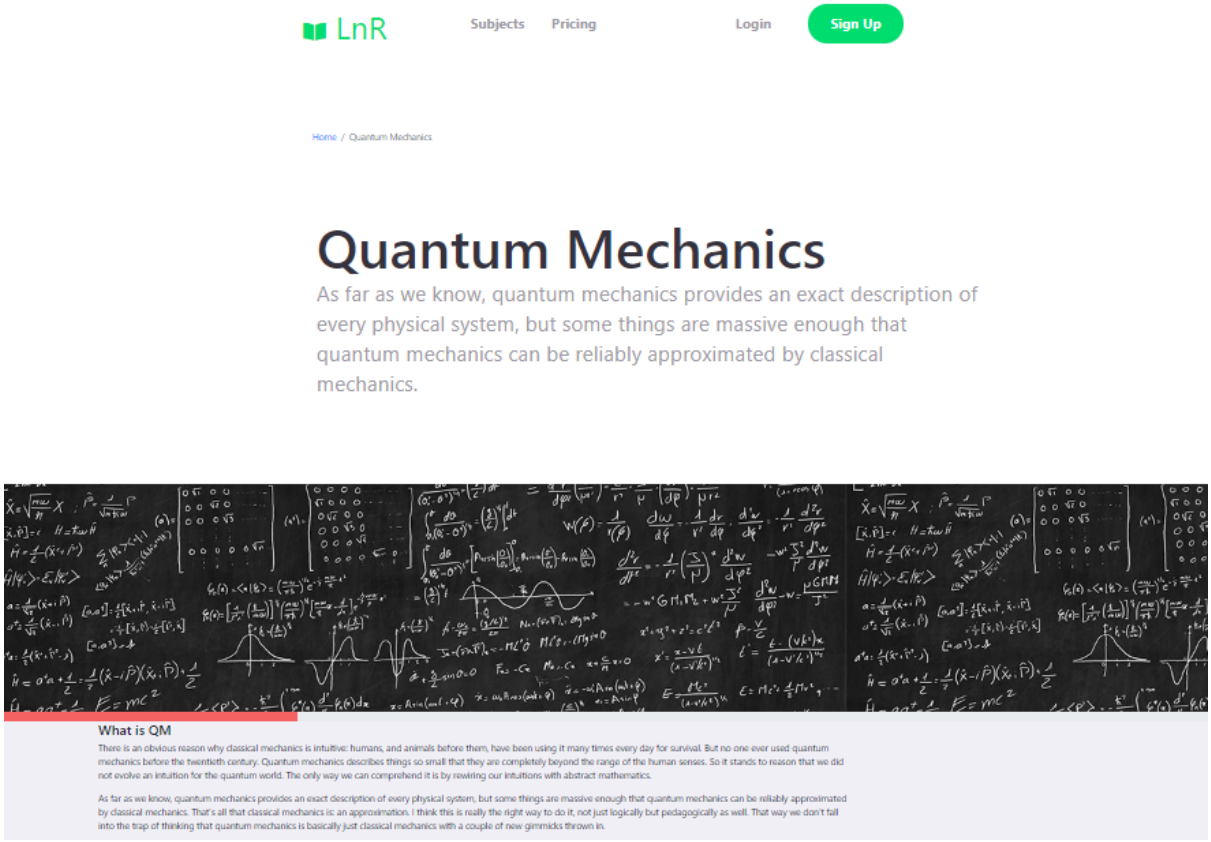
- The service I use for deployment was GitHub. The last step was a series of tweaks to the final website based on feedback I received.

Screenshots from the final website

Home page


LnR
Subjects
Pricing
Login
Sign Up
Science Education For Curious Young Minds
Tackle science topics on the frontier, informed by the most current research, years before it appears in mainstream textbooks.
Get Learning
Enter Email
Send Email
Advance Your Horizons
Here are some of the features that LnR has to offer.

Lesson Page



3. Conclusions

- I achieved in creating a science education website that targets a teenage audience.
- The finished site provides a good basis and framework that I can later extend into a full working website if I choose.

- **Shortcomings**
- Given the visionary nature of this project, there were a few originally planned features that, for many reasons, failed to make it into the final site.
- A few features were impossible to implement without also doing the **back-end** server coding. These included working sign-up and login, session progress storage. At the time, I simply lacked the skills successfully implement these features.
- The audio player, such a central feature during the design stage, mostly fell to the wayside during production. Successful execution would have made the final website too memory intensive, requiring too much space. In the

end, I focused more on using the illustrations and images to achieve the original goals.

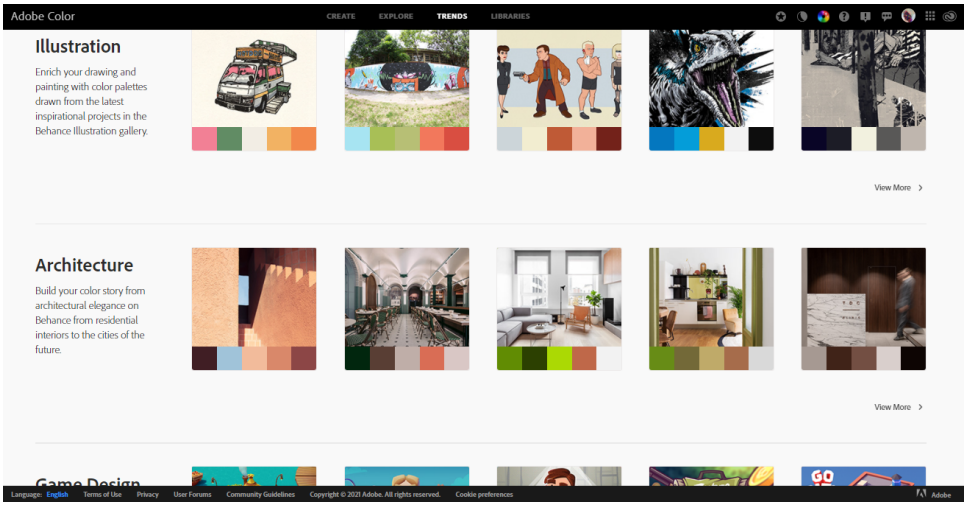
Improvements

- Reflecting on this project, there are some things I would have done differently if I had to to do this project again. I would have kept in mind the very short production **time frame** of the project. This would allowed me plan the project better, avoiding the elaborate plans for various features that ultimately couldn't be implemented.

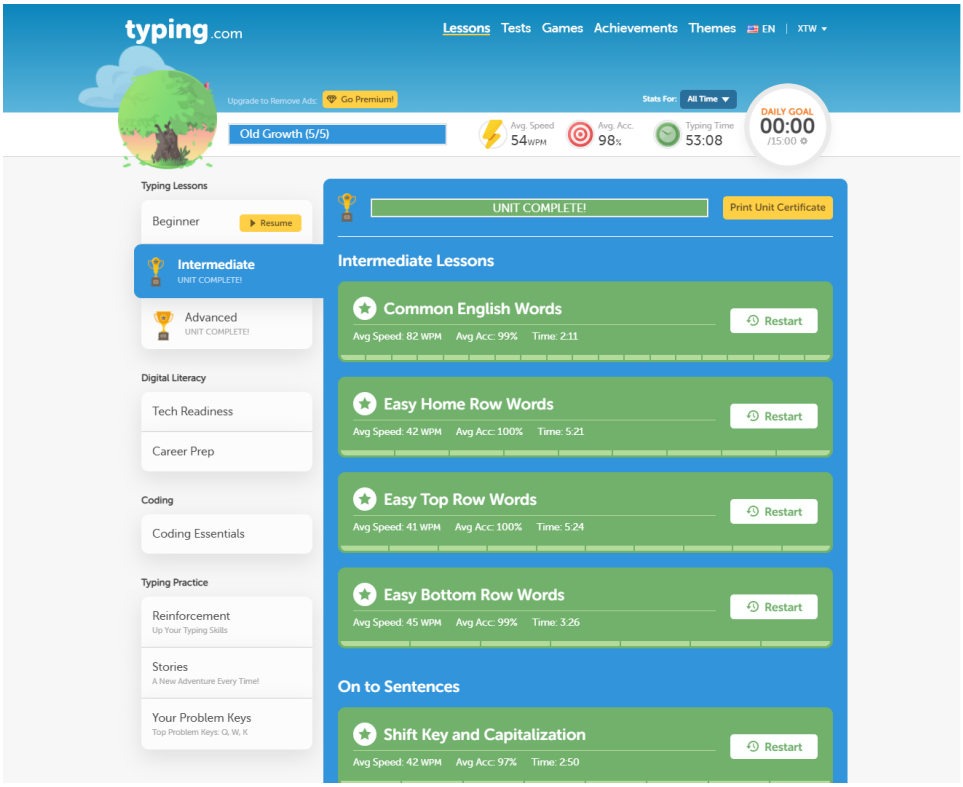
4. References

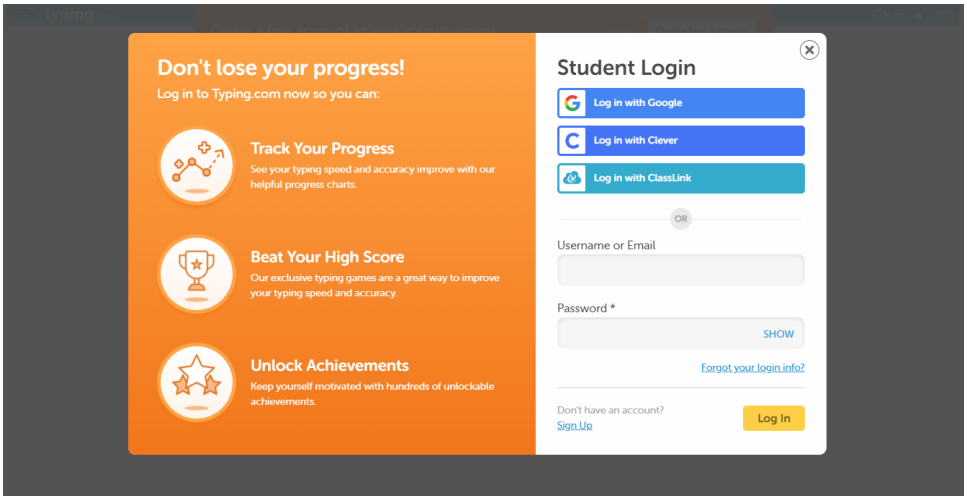
- Eloquent JavaScript
- Sean Carroll: Something Deeply hidden book
- https://www.w3schools.com/: For information on how to implement difficult components (e.g. parallax, audio player, progress bar)

▼ Colour scheme. https://color.adobe.com/trends

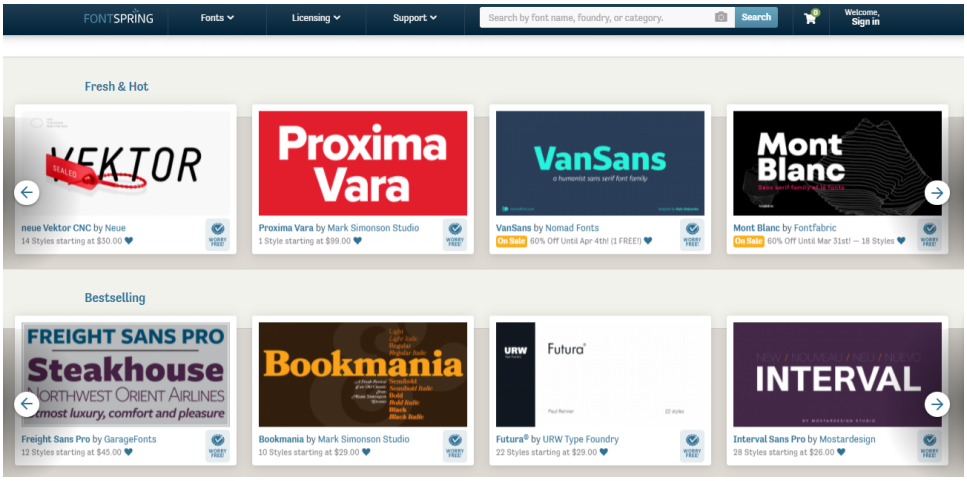


▼ Visual layout inspiration: https://www.typing.com/student/lessons





▼ fonts. https://www.fontspring.com/



https://www.youworkforthem.com/