Introduction

In this report, I will be describing what my project is, showing my planning, design of my coursework for the COMP1004 module and the eventual result. The SPA project I've chosen is a typing test website, like available resources such as Monkeytype.com or typingtest.com.

Project Vision

The reason I am making this project beyond for a university project is tied to my personal goal to increase my typing since 2020, which lead to me learning how to use all ten of my fingers and applying constant practice to get the muscle memory.

The intended users for this project would be employers and learners. For learners, this could be an effective platform to practice your typing skills, with the stats giving feedback on the user's progress, giving them some feedback to see how well they're doing and how effective it is. By extension, employers could use this a way to test potential hirers' typing test within an interview or a small quick external test. If they're a user of this website beforehand and they've effectively practiced their typing skills, employer will view this positively and more likely to higher them.

This would be important in office jobs or stenographers where typing speed is crucial skill as it impacts efficiency of job performance in these types of jobs and tend to improve collaboration between team members, as communication tends to be written channels over email or similar ways and properly conveying their thoughts and needs quickly and properly results in improved teamwork and quickens the work efficiency. In the cases where accuracy is necessary, the website also fulfils this purpose as it does not accept a word be submitted until the word is correct, ensuring that if a mistake is made, you must go back and fix it. This means that in more administrative tasks such as filing databases or forms or customer service tasks, work can be far quicker and far more accurately with no extra costs.

Ethical, Legal and Social Considerations

This is limited due to the scope of my project being a university project but this that may want to consider is the legal concerns around the storage of account information. With my website, it uses Local storage which would be improper for an actual website due to its weak security. Although, the data stored is rather limited in importance as no personal data is stored perhaps only the password as users often reuse passwords. To mitigate this even a bit, hashing and strong passwords practices are enforced to ensure the user has some security.

There are a few social considerations to consider due to the harmless nature of this website but if the project would be introduced into the public, there might be some accessibility concerns, mainly adding a high contrast mode and other customization options for those with disabilities. In addition, it would be useful to add a way to send some feedback on the website as a social feature to allow it to grow more effectively.

SDLC

The software development lifecycle is the industry standard of how software applications are made as it brings an appropriate structure to ensure quality software is created. It brings the massive task of creating new software into smaller manageable tasks making development easier and ensure any time-constraints and budget constraints are worked around, as needed resources become better estimated and established as the development continues.

There are 7 stages of the SDLC:

- Requirements Analysis
  - Identifying features of software
  - Shown by user stories.
- Design/Planning
  - Design software based on requirements.
  - Planning Timeline
  - Shown by sprints.
- Implementation
  - Coding project
  - Shown by sprints + GitHub page.
- Testing
  - Testing features seeing if code is correct.
  - On testing.docx
- Deployment
  - Releasing software
- Maintenance
  - Continuing working on the program, removing newly found bugs and add/changes features.

To apply this lifecycle, I used the Scrum implementation of agile model and due to the nature of this project this remained with the first 4 stages. I started with my requirement analysis once I figured out my idea of my typing test website, formulating them into user stories and outlining a general idea on what each feature would do. This then developed in to use case scenarios of how features would interact.

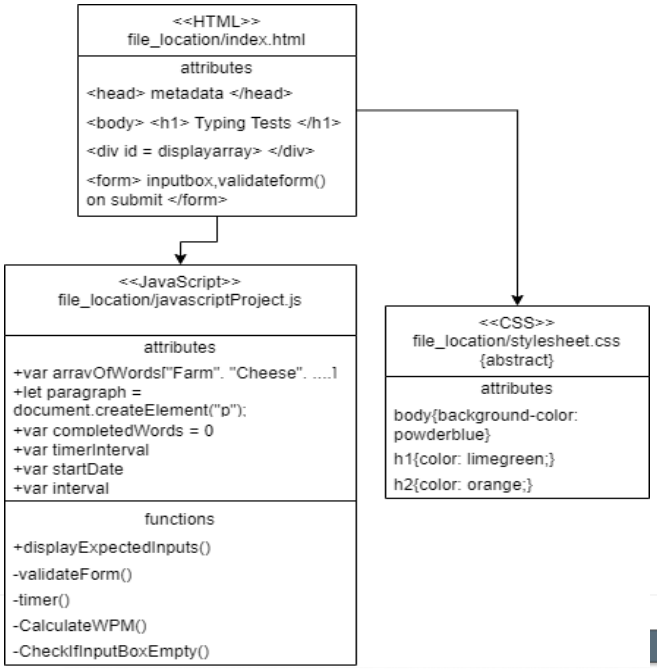
User Stories

Use case scenarios.

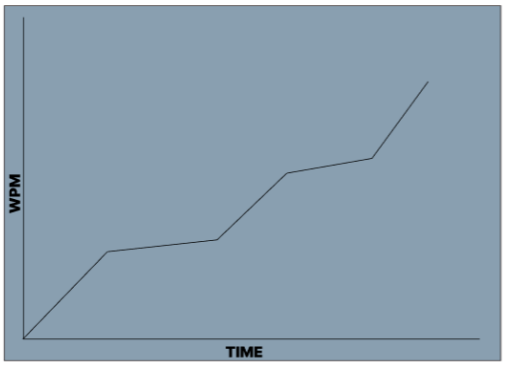
Name	Measure WPM
Short Description	Start timer and count number of written words
Precondition	User has begun typing
Post Condition	Typing test ended
Error Situations	User refreshes page or go to login/sign in/stats page
System state in the event of an error	Typing test is ended (alert is sent before the change unless refreshing page)
Actors	User
Triggers	Desire to do a typing test.
Standard Process	1. User Logins/Signs up 2. User decides difficulty options 3. User start typing test 4. System starts timer. Typing test 4. User inputs words as needed to get as high as possible WPM 5. System ends typing test as timer reaches set time limit 6. System sends alert 7. WPM is calculated and shown on the left
Alternative Process	1' User starts typing test

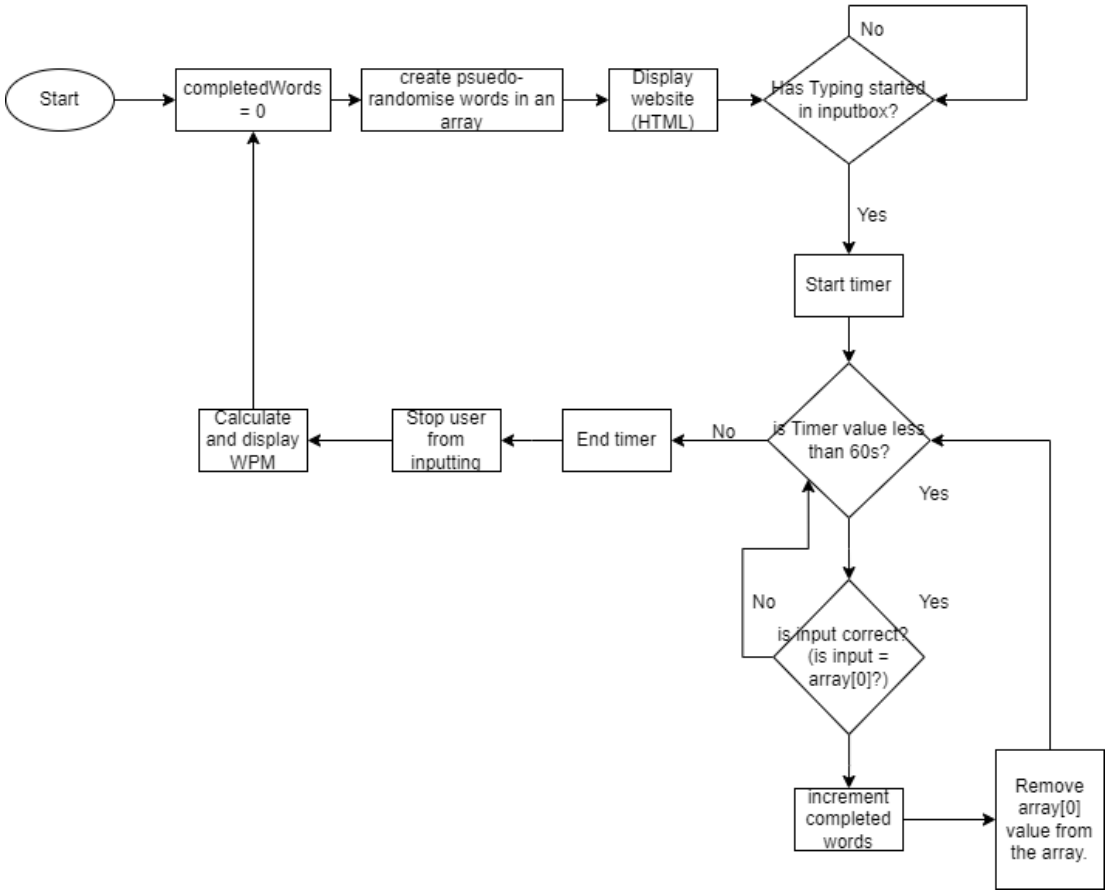
Name	Viewing progress
Short Description	Viewing stats
Precondition	User wants to see stats
Post Condition	User stats are shown
Error Situations	User refreshes page or go's back
System state in the event of an error	Typing test page is shown and if refresh, user is logged out
Actors	User
Triggers	Desire to see stats.
Standard Process	1. User Logins/Signs up 2. User clicks stats button 3. Stats of that user is shown - May be empty if new user.

Next was the design phase, as during this time as I created creating the intial protoype design of my website and how it could look like, which is below, and learning the main features of the langangues used in this project (HTML, CSS and JavaScript).



Wireframe - 2





The implementation phase began as the second sprint, creating an intial prototype with the majority of the main features and continued throughout the sprints. This is the same for testing phase and userfeedback gathering, ensuring features worked proficiently. When new features were being worked on, the design phase was continued to be implemented, adding to the ULM and wireframes when necessary, developing it over time. This is avaliable on my github page, within ‘design’ folder.

Sprints

This table is representative of the status of each of my user stories features comparative to my sprints.

Features	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8
Able to measure WPM with a test		WPM/Timer visible in console, array not randomised	WPM/Timer displayed on website; array randomised and tested	Gathered user feedback, attempted word generation, failed.	Tested and implemented user feedback implemented, with positive reception			
Sign in/Log in					Created HTML of login page, investigated JSON	Log in works after fetching data from JSON	Able to create an account which persists.	Tested.
Store stats and view them							Stats added to JSON, can be viewed with graph included.	Tested.
Export stats into JSON							Implemented and tested.	
Difficulty options				Timer options implemented	Difficulty options implemented	Difficulty options tested (fixed bugs)		
Displaying correct/incorrect inputs				Feature implemented and tested..				
Security features						Passwords hashed, Passwords standards enforced	Tested.	
UI	Wireframe created		Wireframe implemented	Took user feedback	Wireframe redone, dark mode and light mode implemented.	Wireframe created for login/signup pages and implemented	Created wireframe for stats page and implemented.	Tested.

Not worked on	Started but limited progress (<50%)	Over halfway done (>50%)	Nearly finished, if not finished, testing required. (>85%)	Finished (100%)

The next page is my backlog and goals for each sprint week by week as I filled them out.

The next page is my sprints, detailing my backlog and what I'm doing in the week.

SPRINT WEEK 1-2 (1)	SPRINT WEEK 3-4 (2)	SPRINT WEEK 4-6 (3)	SPRINT WEEK 7-8 (4)
Goals -Outline requirements in user stories - Create a flowchart to display the logic of website (basic logic to be used in prototype) - Create a wireframe using Figma (UI) -Create UML for prototype logic -Learn CSS/JS/HTML	**Goals** - Create a skeleton of the code - To amend ULM + Flowchart to match actual logic if any mistakes - Adhere to wireframe - Create a working prototype • Can do a 60second typing test for array of words. • Gets WPM (words per minute)	**Goals** - Create new ULM to show new structure - Redo UI to fit the wireframe - Display WPM + timer onto website rather than console - Make correctly inputted words autodelete on submit - Creation of proper testing - randomised input array generation	**Goals** - Implement difficulty options - Implement timer options - Creation of DOM diagram - Tell user typing test is over - Take some user feedback - Display errors on text live - Change WPM calculation - Put comments in .js file for better readability - Attempt to use Word generation instead of using a randomised array (abandoned feature)
Status - Completed - Goals met	**Status** - Completed - Goals not met	**Status** - Completed - Goals met	**Status** - Completed - Goals not met
Next steps - Using UML and flowchart as base create a working prototype Features of prototype -timer -calculation of wpm - typing the required words in an input box	**Next steps** - Using UML and flowchart as base create a working prototype Features of prototype -timer -calculation of wpm - typing the required words in an input box	**Next Steps** - Implementation of difficulty/timer options - Letters go red if incorrect, or green if correct - Creation of new Ulm to show structure - Gather user feedback from close friends	**Next Steps** - Creation of ULM to show new structure - Show WPM live - Make login system - Implement user feedback - Attempt to do word generation, limited success, hard to make genuine words
SPRINT WEEK 9-10 (5)	**SPRINT WEEK 11-12 (6)**	**SPRINT WEEK 13-14 (7)**	**SPRINT WEEK 15-16 (8)**
Goals - Create ULM to show new structure - Show live wpm and display it (abandoned feature) - Make login system html, allowing user to go back -Create new wireframe - Clean UI up further (user feedback) - Once implemented, take more user feedback - implement a light/dark mode for UI - Implement difficulty options effectively - update User stories with security features	**Goals** - Allow to sign in with their details - To be able to read from a Json file (holds passwords/users) - Allow user to create new account and store it in Json file - Allow password to be shown/hidden - Allow creation of new user - Ensure commenting has continued in new code - Create wireframe log in/sign in/log out html additions and implement them - Hash passwords for security - Fix paragraphs not deleting properly bug - Added security to user stories	**Goals** - Allow WPM to be stored and assigned to their user details - Put theses stats within a Json that can be exported, using blob - Create wireframe for page which displays theses stats - Allow stats to be visible on webpage - Replace alerts with nice UI implementation (keeping alerts) - Style log in/sign in/log out html additions according to wireframe - Allow WPM data to be displayed as a graph (using chart.js) - Retest old features and add new features to testing	**Next Steps** - Final ULM - Create Sitemap - Create Packet Diagram - Writing Report - Add any more QOL features if desired, e.g. more stats in graph, ability to change username/password, ability to reset stats, etc. - Do final testing to ensure all features are up to standard - Get user feedback and fix any found errors - Delete account button to show CRUD
Status - Complete - Goals not met	**Status** - Complete - Goals not met, not possible to meet unless using node.js	**Status** - Complete - Goals not met	**Status** - Incomplete - Main goals met
Next Steps - Word/paragraph generation or grabbing it from the internet seems practical for a larger project but not needed for my scope. - Allow to sign in with their details - Allow WPM to be stored and assigned to their user details - Allow it to be displayed as a graph	**Next Steps** - Allow WPM to be stored and assigned to their user details - Put theses stats within a Json that can be exported. Using blob - Replace alerts with nice UI implementation - style log in/sign in/log out html additions - Allow WPM data to be displayed as a graph	**Next Steps** - Final ULM - Writing Report - Add any more features if wanted, e.g. more stats in graph, ability to change username/password, ability to reset stats etc	**Next Steps** - Submission

Gathered user feedback:

User feedback (1.1)

- Taken 4th sprint.
- Found it abrupt that words disappeared after every word, making typing harder as not as natural to do.
- UI could be cleaner.

User feedback (1.2)

- Taken 5th sprint.
- This was taken after user feedback initially implemented.
- Like the change that words are deleted at end of line
- User likes it far better, flows far nicer and UI is far cleaner.
- Only suggested slightly darker shades for background colour
- Suggested that background dimmed if typing test begins, like a focus mode.

User feedback (1.3)

- Taken at the end of the sprint cycles.
- Well, coloured for both dark and light mode.
- User found bug with starting a new test as a newly created signed in user after finishing the first one.
    - Fixed shortly after
- Very responsive
- Easy on the eyes and intuitive to use.
- Buttons colouring slightly out of style, stands out.
- Would be nice to have button to change light/dark mode instead of using system settings.
- Appreciated that word must be correct before continuing to next word.

User feedback (2.1)

- Taken 4th sprint.
- Implemented CSS does not work on their laptop for some unknown reason.
- Potentially OS issue as they're using windows 10, not windows 11 but can't say for sure.
- Used on my device, User had similar feedback to 1.1.

User feedback (2.2)

- Taken 7th sprint.
- This was taken much later than 1.2 due to timing.
- Appreciated the changes.
- Desired for more stats data
    - Too little time to implement more stats data.

User feedback (3.1)

- Taken at the end of the sprints cycle, though seen it throughout as user been helpful in some of my coding challenges as they're experienced with theses languages.
- Website is done well, dark mode is done well, looks clean.
- Not suitable for all resolutions/screen widths and/or heights not ideal
- Found visual bug, in dark mode, typing turns the text black which blends in with the background.
    - This has been fixed.
- Suggests any extra features would be great to add but not necessary for the scope of this project (such as more stats, able to change password etc)

User feedback (4.1)

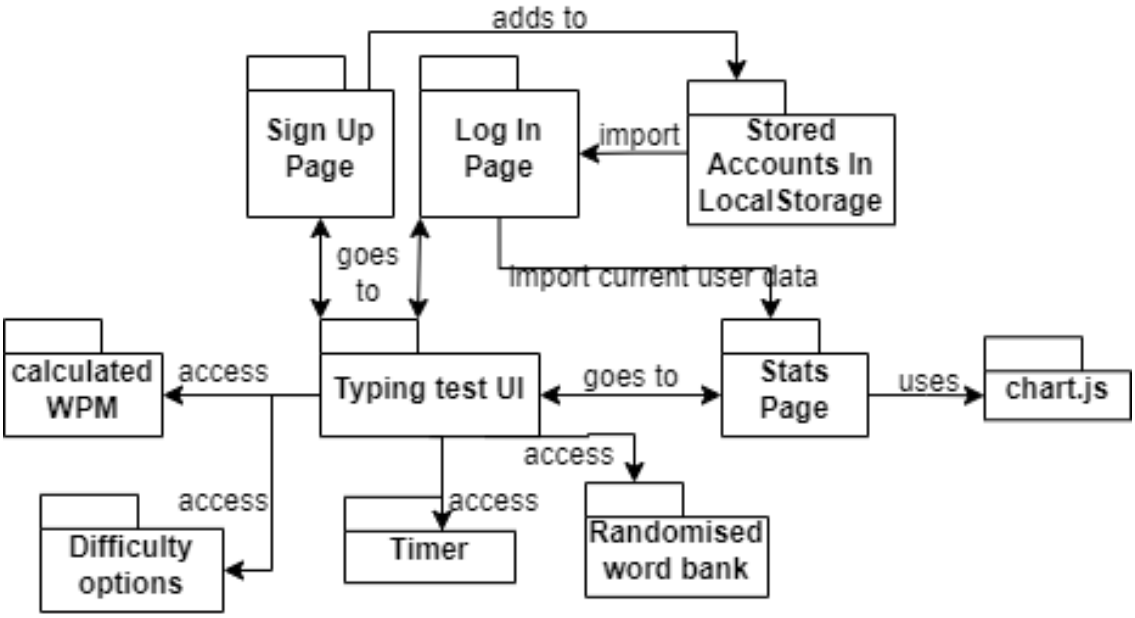
- Taken at the end of the cycle.
- Disliked the visual of shifting words up at end of line, leading to mistyping, though minor issue and personal preference.
- Appreciated the styling showing correct/incorrect.
- Overall, decent styling and functions well

Testing of each of my features has been implemented, shown in testing.docx (within coursework submission folder) on the GitHub, done by me and the users who gave user feedback as primarily they were used to help improve user experience and find any errors they could. Errors were fixed once caught and user feedback implemented and retaken to see how effective it was. This was true unless user feedback was taken at the end of the sprint cycle (8^{th} sprint) as at this stage my project is finalised.

Architecture

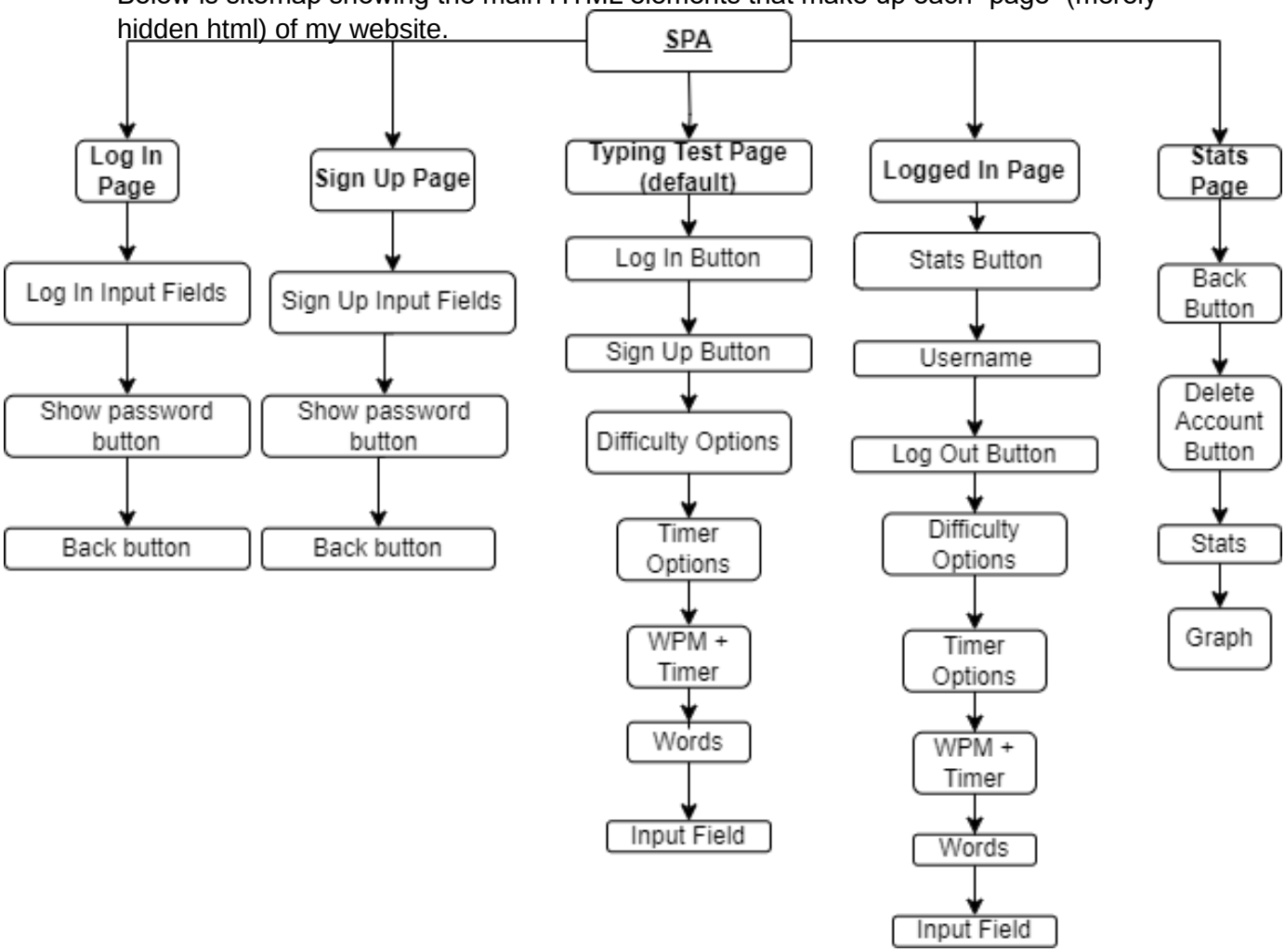
These are the diagrams and final versions of all my ULM diagrams of my website. Here is a packet diagram, showing how my code interacts with one another as a whole.

Packet Diagram



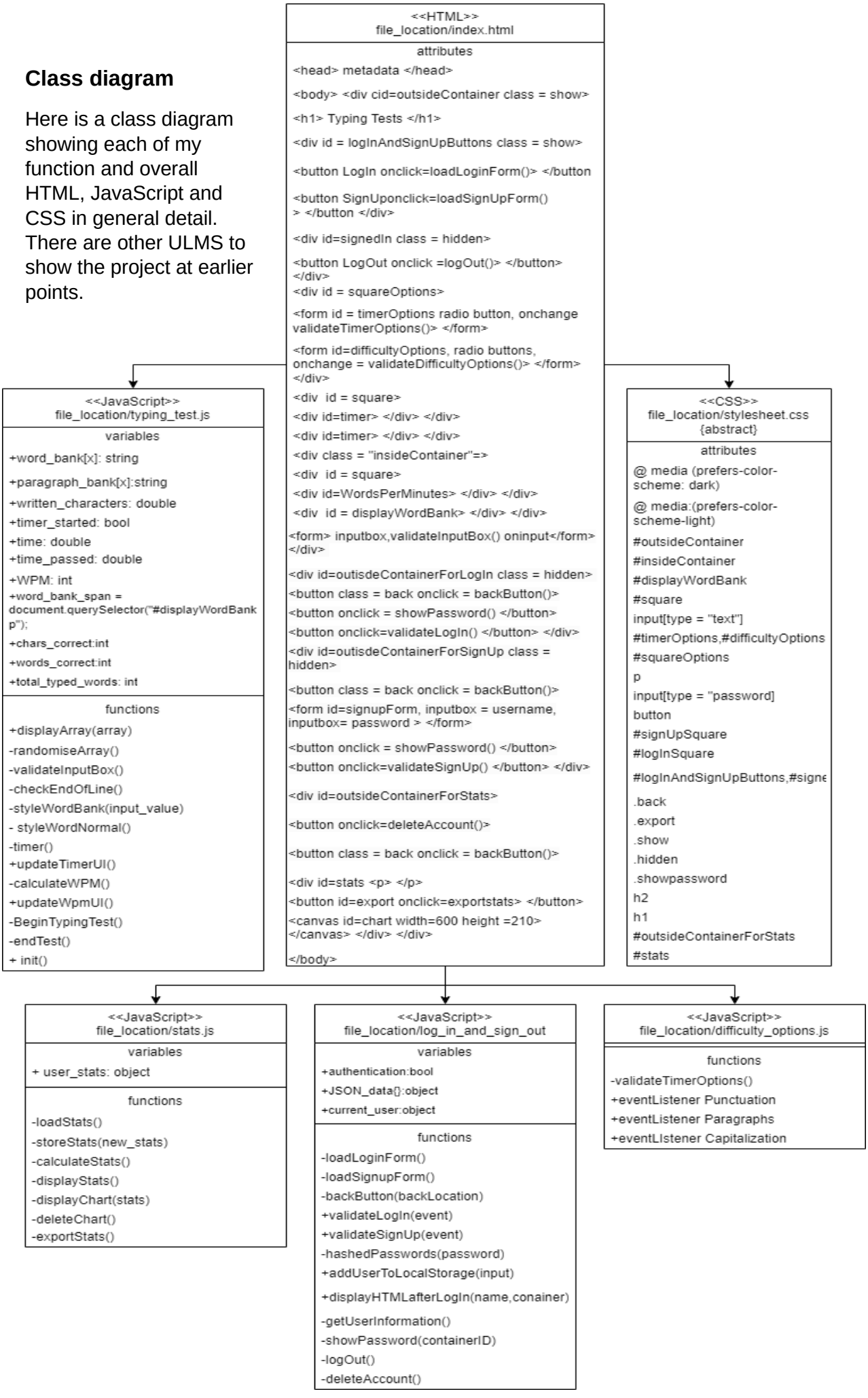
Sitemap

Below is sitemap showing the main HTML elements that make up each “page” (merely hidden html) of my website.

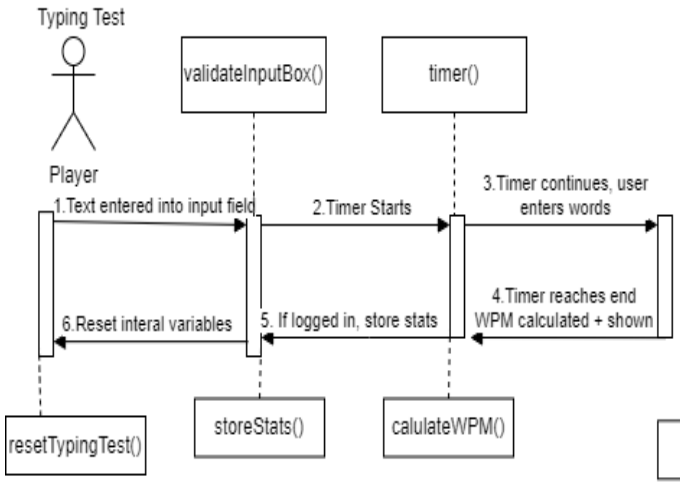


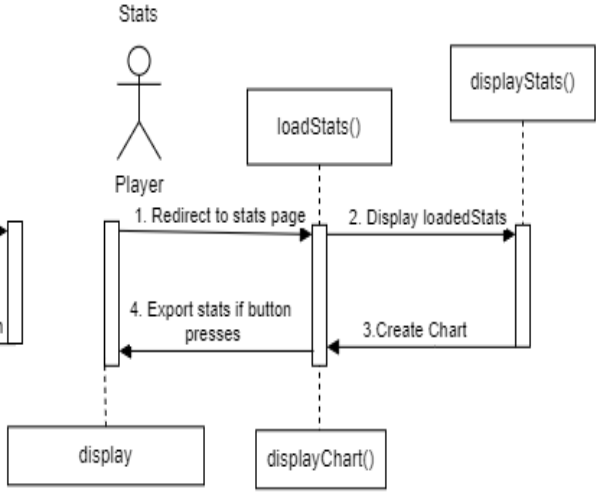
Class diagram

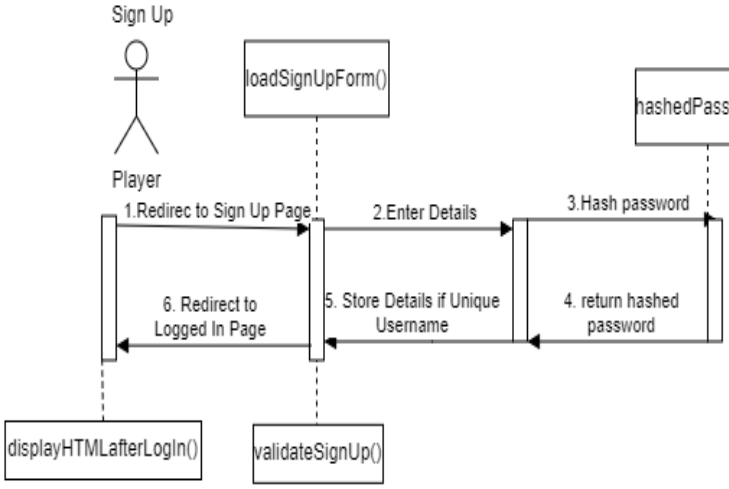
Here is a class diagram showing each of my function and overall HTML, JavaScript and CSS in general detail. There are other ULMS to show the project at earlier points.

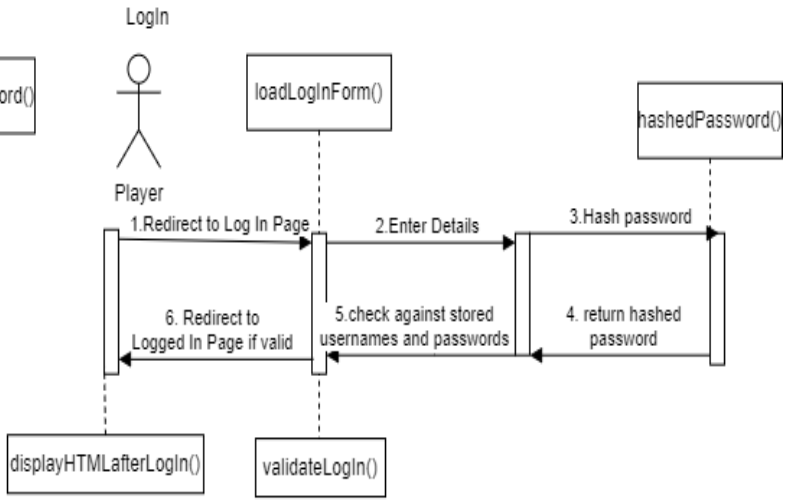


Sequence Diagrams





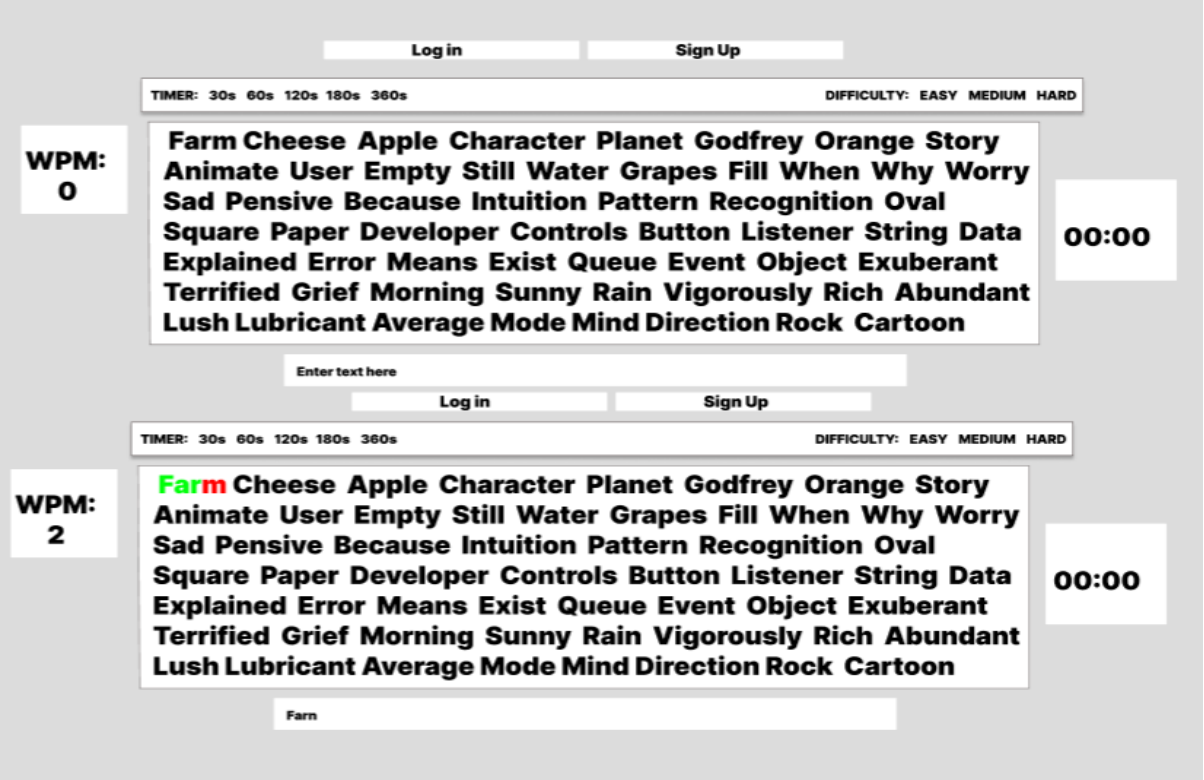




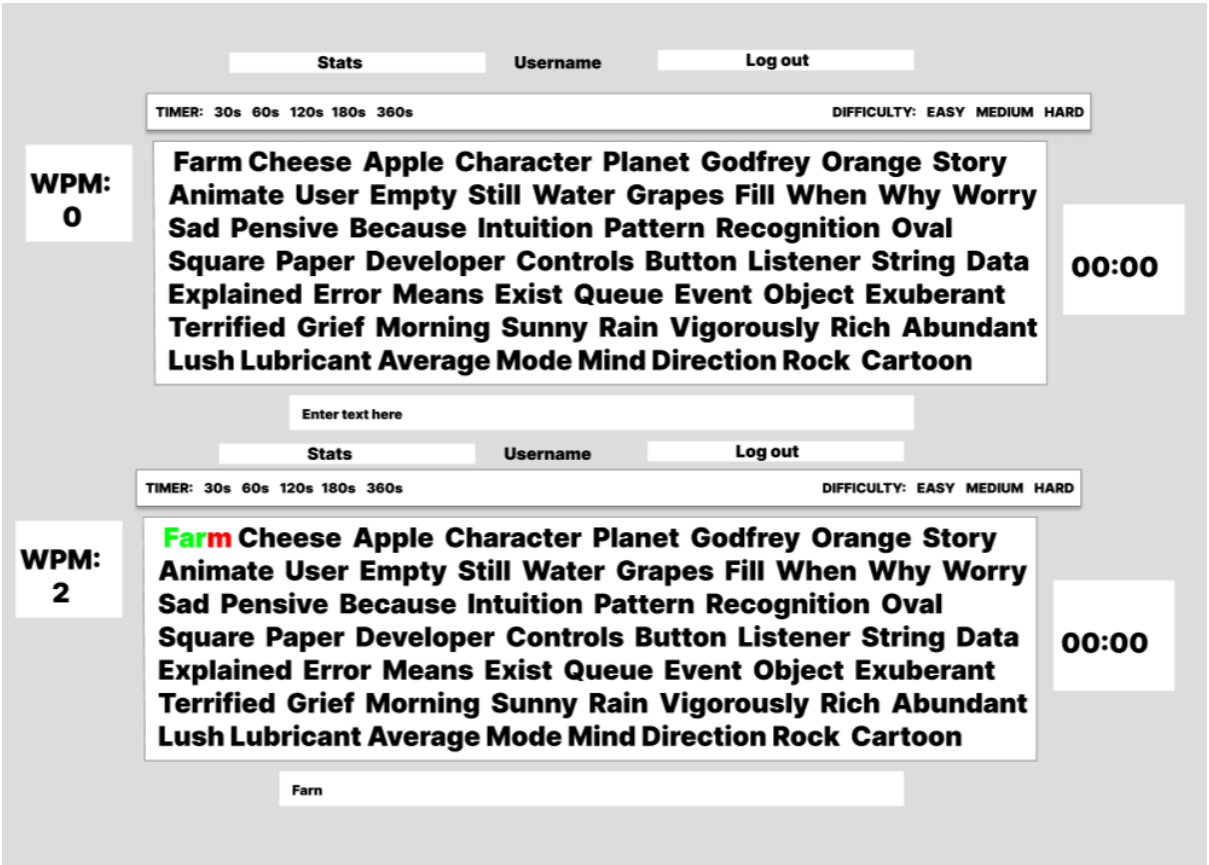
Wireframes

Light Mode

Default Typing Test Page



Logged In Page



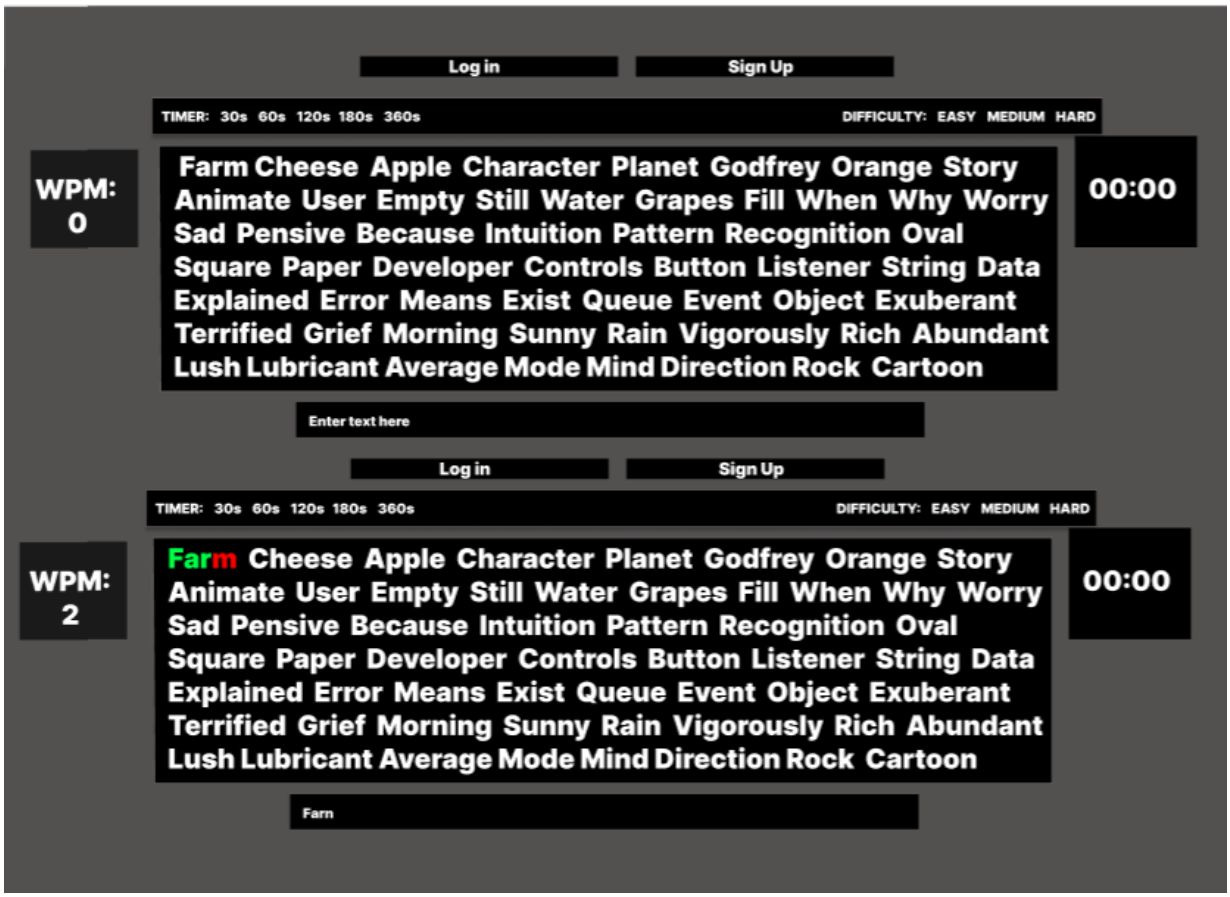
Stats Page

Note: delete account button not shown in wireframe

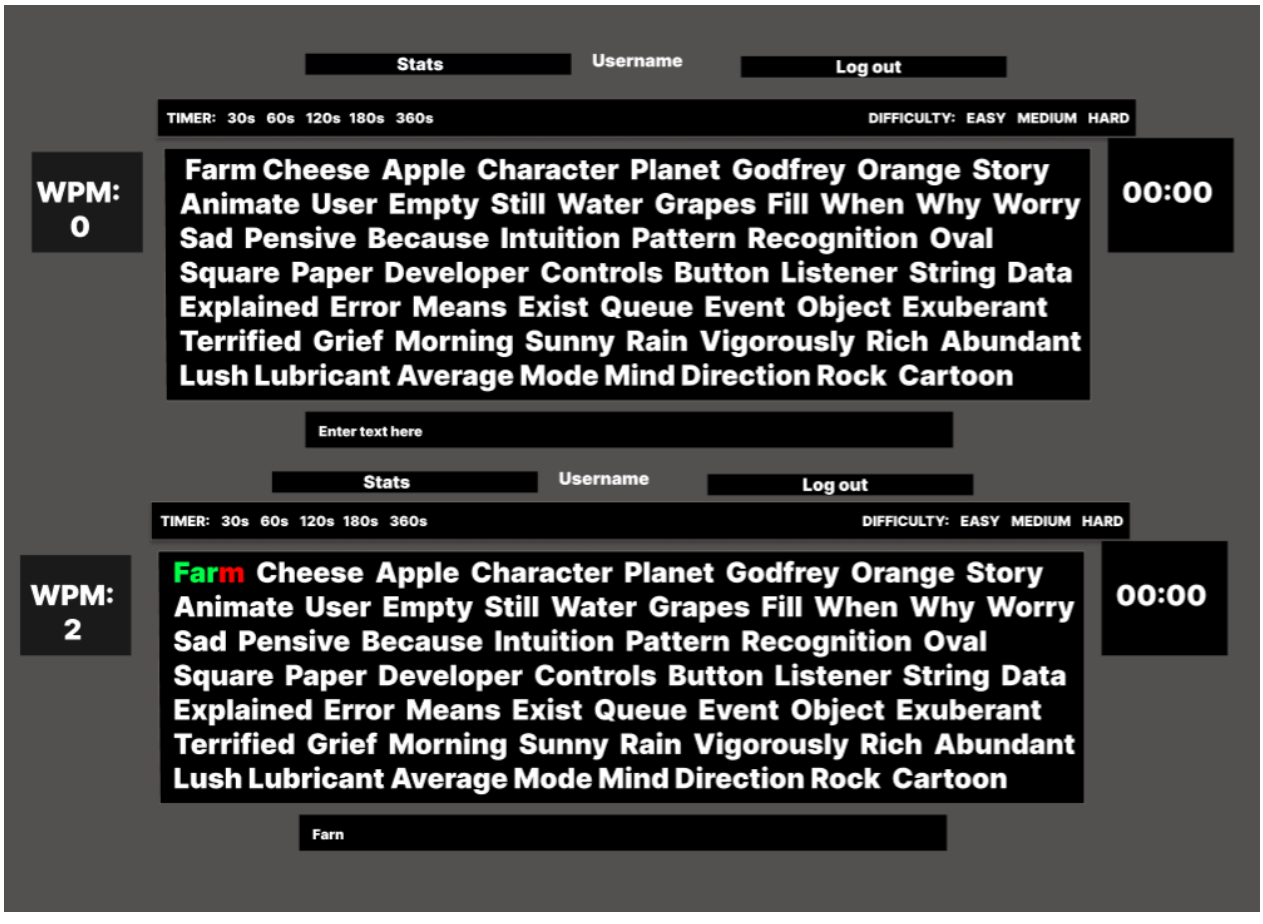
Login/Sign Up Page

Dark mode

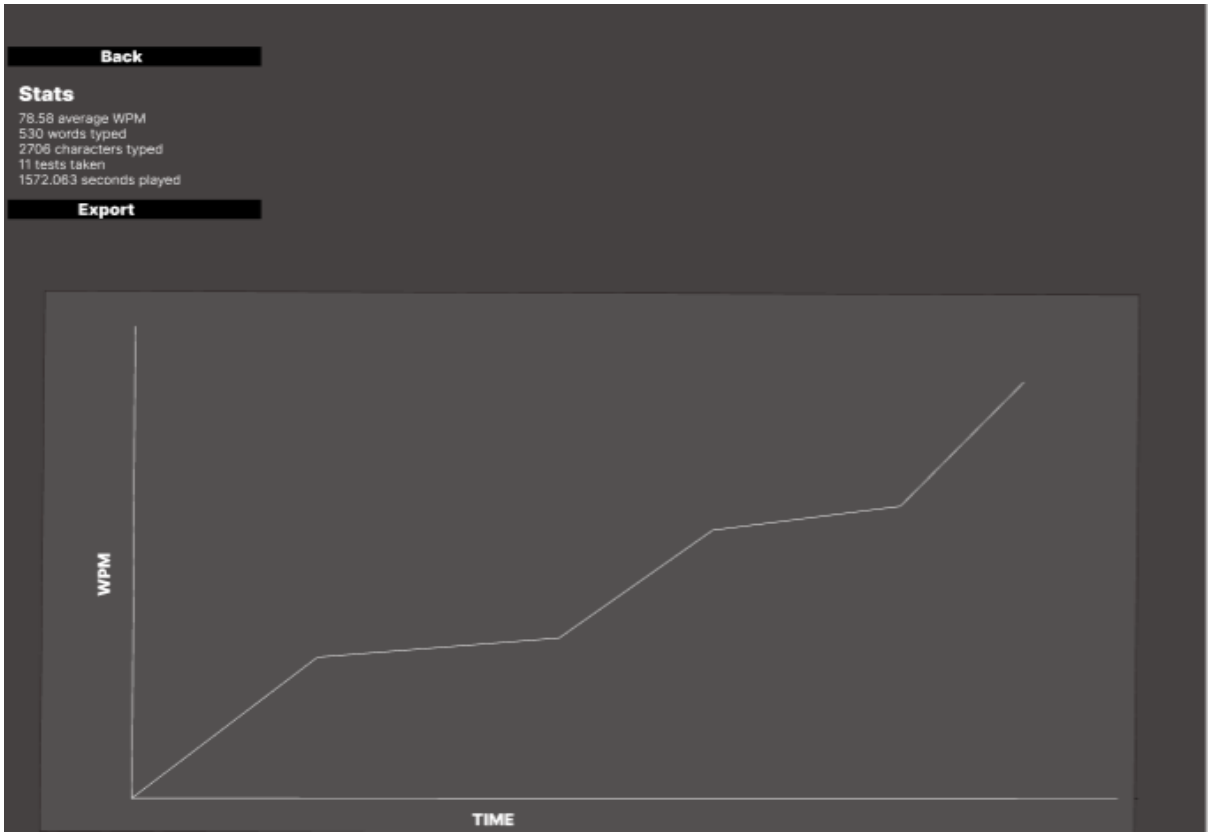
Default Typing Test Page



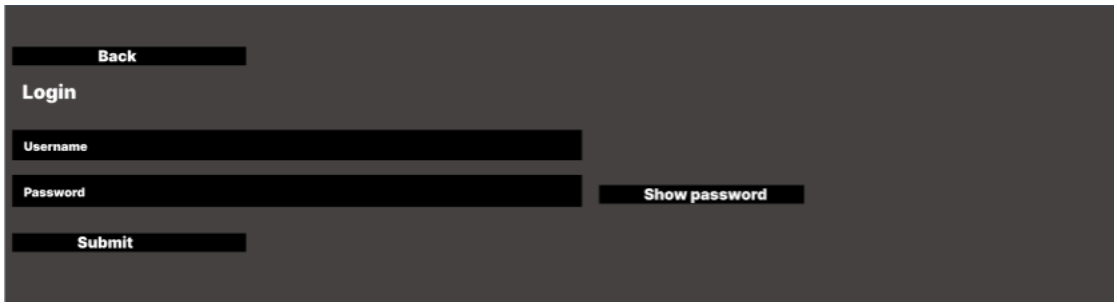
Logged In Page

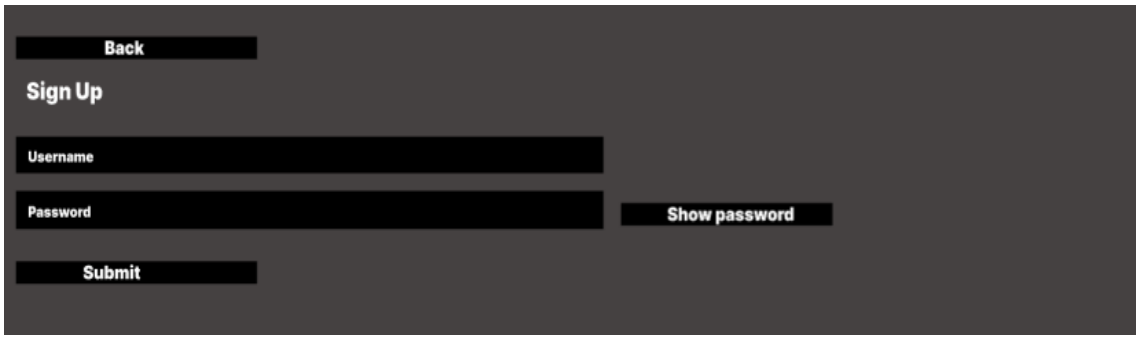


Stats Page



Log in/Sign Up page.





Conclusion

Challenges and constraints faced during creation process.

My primary challenge when doing my project is caused by coding as consequence of my lack of experience with the three languages we have to use: HTML, JavaScript, and CSS. HTML and CSS weren't too difficult due to their rather simpler nature, but JavaScript took significant time to become component at. While the initial learning curve weren't too high due to its sharing qualities with both C++ and python, my primary coding languages.

What was the initial challenge was understanding how to effectively incorporate the JavaScript into my HTML, so it responds to my inputs using document.getElementById, the "onclick" / "oninput" attributes and event listeners. Once I understood that, progress was steady.

This was halted in the fourth sprint when I attempted to style the words and reset words at the end of the line. This initially challenged me as I need to style each letter individually leading to the discovery of spans. Although, it wasn't' that simple. It required me to engage with new concepts such as query selector and nth child. This was arguably the hardest part of the coding for this project, requiring some ingenuity to solve the problems.

Next challenge of my coding adventure was inputting and exporting to a JSON which wasn't hard by itself, but the hardest part was finding out the necessary information to apply it without using node.js as node.js seems so efficient as it that very few people do it. This required some help of the lectures in the lab sessions as other also struggled to solution, including the lecturers. This encouraged me to change my IDE from notepad++ to VS code as an extension called Live Preview circumvented this issue by hosting a local web server when displaying the website which is required to use fetch when getting data from the JSON. This is arguably the biggest constraint of my project, as without this or using the included bat file, it is impossible to login/signup.

A general challenge with this project was juggling it with other university projects and responsibilities, although using the labs, I found it simpler to take those 4hours and work proficiently as much as I can during that time, getting lot of it done early.

Reflection

Overall, I am quite happy with my project as I filled out all my initial goals and created some new ones along the way such as the security features. I wished I had more time or technical expertise in node.js to properly implement the features as initially intended. I feel like I properly did the sprints and my backlog never felt overwhelming or unmanageable to fulfil. On the other hand, it can be argued my sprints were simplistic in nature and I could have achieved more but that does not take in account of the learning process that was going on behind the scenes of this entire process. I would have liked more user feedback as I found that particularly helpful in my process of creating as it allowed me to get a new perspective on how to change improve and change things.

GitHub Repo link

Repo: https://github.com/MxFrgsn/COMP1004-Project

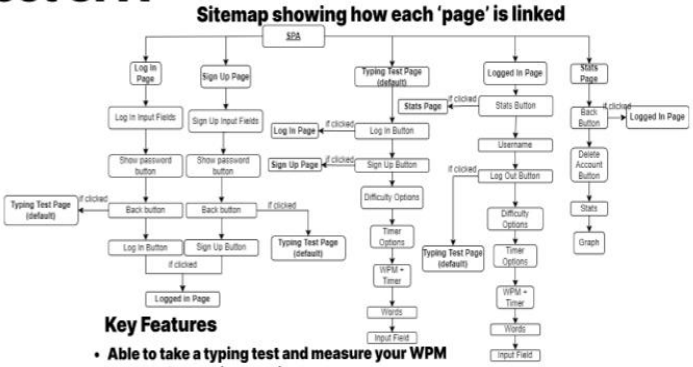
Poster

Typing Test SPA

Project Vision

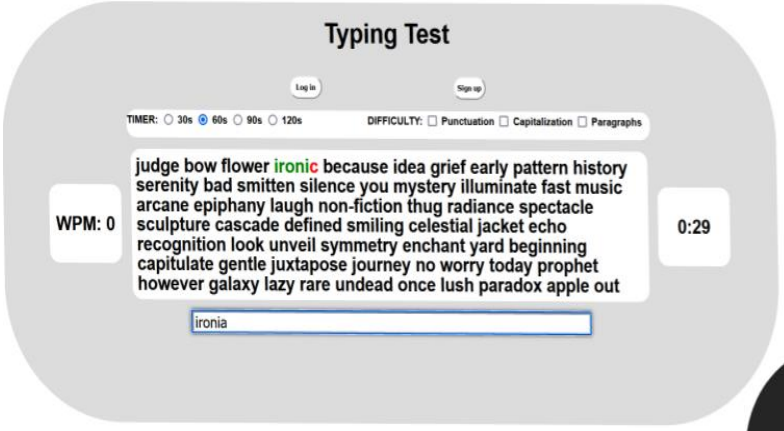
- Personal interest in improving typing ability since 2020
- Intended for learners such as myself
- A practical way to do consistent practice and monitor progress
- Useful skill in office jobs jobs
- Can also be used by employer to test potential hires

Sitemap showing how each 'page' is linked



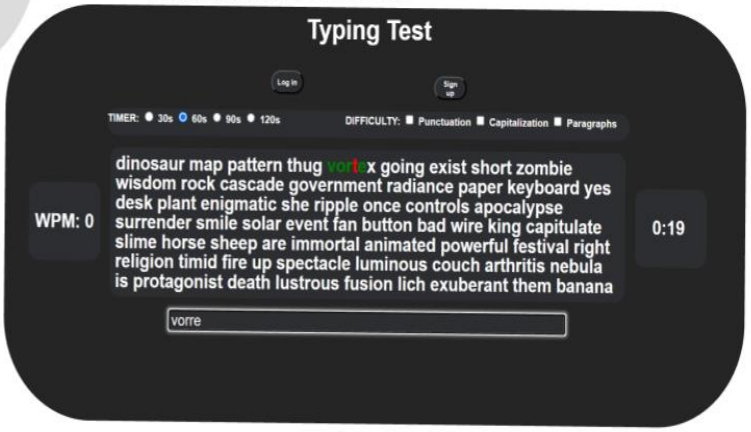
Key Features

- Able to take a typing test and measure your WPM
- Able to change timer options
- Able to change difficulty options (punctuation, capitalization, paragraphs)
- Able to sign up
- Able to log in
- Able to store stats information and view it
- Hashing passwords for security
- Strong password enforced
- Able to delete account
- Able to download stats information



Challenges

- General challenge with learning HTML, JavaScript and HTML as its all new to me
- JavaScript has been causing the most issues (being 80% of my SPA)
- Particular issues include:
  - Styling words showing correct/incorrect inputs
  - Checking if at end of the line
  - Fetching data from JSON without node.js
- CSS of website, graphic design is not my strong point
- How to maintain a SPA feature but still show new pages
- Finding and fixing bugs



Max Ferguson
https://github.com/MxFrgsn/
max.ferguson@students.plymouth.ac.uk