TypingTest

NoorMohammad Alizadeh – CIN4A

ETML – Ecole des Métiers Lausanne

Lundi 09 mai 2022 – Mercredi 08 juin 2022

Chef du projet : Raphael Pasche

Experts : Alain Roy, Yves Bertino

Table des matières

[1 Analyse préliminaire 4](#_Toc103963075)

[1.1 Introduction 4](#_Toc103963076)

[1.2 Objectifs 4](#_Toc103963077)

[1.3 Prérequis 4](#_Toc103963078)

[2 Planification Initiale 4](#_Toc103963079)

[3 Analyse/Conception 6](#_Toc103963080)

[3.1 Concept 6](#_Toc103963081)

[3.1.1 Interface 6](#_Toc103963082)

[3.2 Matériel et logiciels à disposition 7](#_Toc103963083)

[3.3 Risques techniques 7](#_Toc103963084)

[3.4 Points à découvrir 7](#_Toc103963085)

[3.5 Stratégie de test 7](#_Toc103963086)

[3.6 Planification détaillée 8](#_Toc103963087)

[3.7 Dossier de conception 11](#_Toc103963088)

[3.7.1 Environnement de développement 11](#_Toc103963089)

[3.7.2 Outil de versioning 11](#_Toc103963090)

[3.7.3 Serveur WEB 11](#_Toc103963091)

[3.7.4 Préparation de matériel 11](#_Toc103963092)

[3.8 Préparation de l’environnement 13](#_Toc103963093)

[4 Réalisation 14](#_Toc103963094)

[4.1 Dossier de Réalisation 14](#_Toc103963095)

[4.1.1 Version des matériels/librairies utilisées 14](#_Toc103963096)

[4.1.2 HTML 14](#_Toc103963097)

[5 Tests 17](#_Toc103963098)

[5.1 Dossier des tests 17](#_Toc103963099)

[6 Conclusion 18](#_Toc103963100)

[6.1 Bilan des fonctionnalités demandées 18](#_Toc103963101)

[6.2 Bilan de la planification 18](#_Toc103963102)

[6.3 Bilan personnel 18](#_Toc103963103)

[7 Divers 18](#_Toc103963104)

[7.1 Journal de travail 18](#_Toc103963105)

[7.2 Table des illustrations 18](#_Toc103963106)

[7.3 Glossaire 18](#_Toc103963107)

[7.4 Bibliographie 19](#_Toc103963108)

[7.5 Webographie 19](#_Toc103963109)

[8 Annexes 20](#_Toc103963110)

# Analyse préliminaire

## Introduction

TypingTest est un site web pour entrainer sa dextérité à la dactylographie. Il y aura deux modes de test différent sur le site, un mode par nombre de mots et le deuxième mode est par compte à rebours.

## Objectifs

1. Les différents modes de test : par nombres de mots, par compte à rebours
2. Un texte à recopier : les mots français et les mots générés avec des caractères aléatoires
3. Un chronomètre qui démarre quand l’utilisateur commence à taper et s’arrête quand il finit de taper ou le temps arrive à zéro dépendant quelle mode j’ai choisi
4. Gestion des utilisateurs à l’aide des cookies

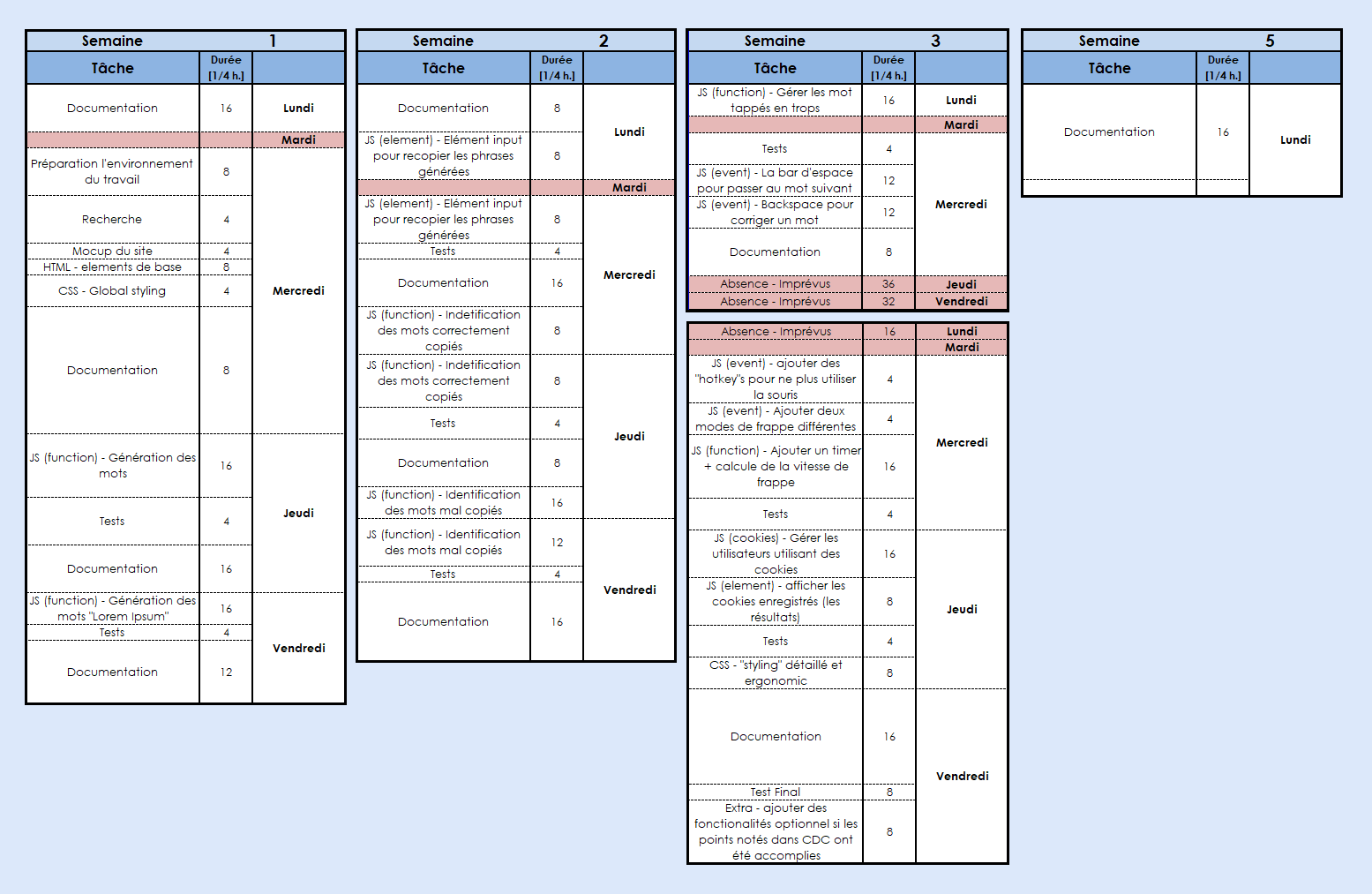
## Prérequis

* Modules ICT orienté WEB
* Usage des Cookies
* JavaScript

# Planification Initiale

La planification initiale a été envoyée le premier jour du TPI aux experts et au chef du projet par email.

Cette planification visualise ce que j’ai globalement prévu comme tâches et me permettra de voir une liste de mes tâches à faire durant ce projet.



**Semaine 4**

Figure - Planification Initiale

**Semaine 4**

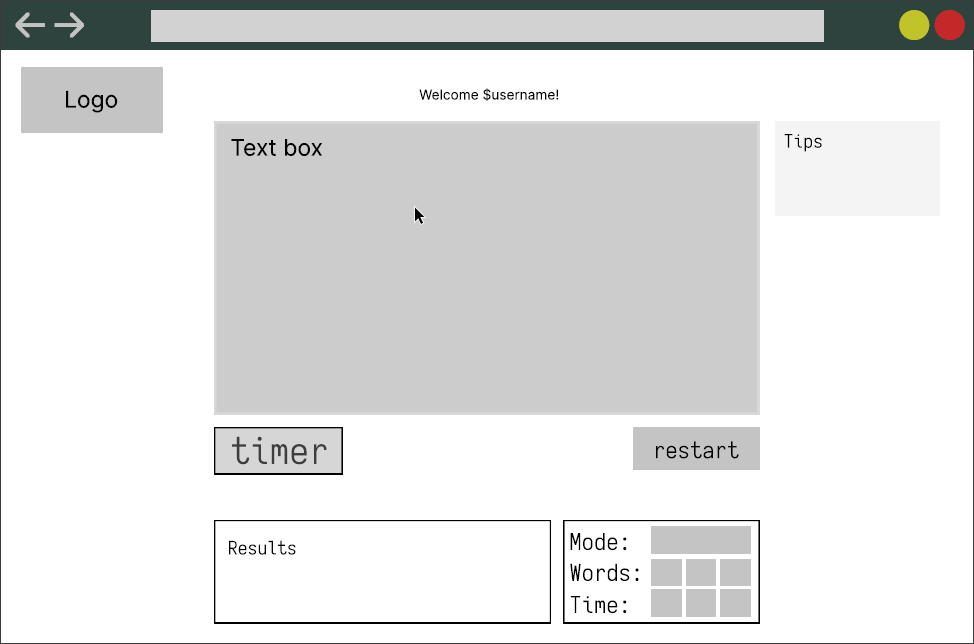
# Analyse/Conception

## Concept

Le plan du site est relativement simple et tout se passe en une page.

### Interface





## Matériel et logiciels à disposition

* Un PC standard de l’ETML (Windows 10)
* Visual Studio Code
* Serveur web sur cloud (Render.com)
* Suite Microsoft Office pour la documentation
* Un dépôt Git privé (GitHub)
* GitHub Desktop

## Les risques techniques

Pendant 4-5 ans de ma présence à l’ETML, le module orienté WEB n’était mon point fort et mon projet est majoritairement basé sur JavaScript, le langage dont nous avons pas appris dans nos modules orienté WEB.

Le développement WEB m’a toujours passioné, même si je n’étais pas assez bien dans le domaine. Durant mon stage d’entreprise au Gymnase de Chamblandes, j’ai pu beaucoup travailler sur JavaScript et m’améliorer.

## Points à découvrir

## Stratégie de test

Les tests seront faits pour les fonctionnalités JavaScript vue les nombres des points à réaliser listés dans le cahier de charges. À la fin de création de chaque fonction, j’effectuerai un test pour valider si je peux passer à la fonctionnalité suivante.

Vous verrez une liste détaillée de mes tests dans le point 5.1.

## Planification détaillée

Figure - Planification détaillée (Semaine 1)



Figure - Planification détaillée (Semaine 2)



Figure - Planification détaillée (Semaine 5)

Figure - Planification détaillée (Semaine 4)

Figure - Planification détaillée (Semaine 3)

## Dossier de conception

Pour ce projet les matériels à dispositions ont été proposé dans cahier de charge et je suis la plupart de ces propositions comme le choix des système d’exploitation car utilisation Windows 10 est plus pratique quand je suis à l’école.

### File, type, vscode Icon in vscodeEnvironnement de développement

Personnellement j’utilise souvent NeoVIM qui est une continuation d’un utile de traitement du texte connue s’appelé VIM. Main pour ce projet j’ai décidé de continuer avec la proposition dans mon cahier de charge. **Visual Studio Code**, car NeoVIM n’est pas un utile optimisé pour être utilisé dans environnement Windows. Une autre raison c’est que je peux aussi utiliser le « Five Server », une extension à VS Code permettant avoir un server PHP en live et se met à jour à chaque fois il y a un changement dans mon code HTML, CSS, JS ou PHP.

Figure - Visual Studio Code

### GitHub Logos and Usage · GitHubOutil de versioning

Normalement j’utilise GitLab parce que il a une meilleur offre gratuite comparant au GitHub **mais** dans ce projet **j’utilisera** **GitHub** comme utile de versioning. Et la raison est seulement que je ne veux pas mélanger mes projets de l’école avec mes projets personnels et je voulais pas créer un nouveau compte GitLab.

Figure – GitHub.com

### Serveur WEB

Dans ce projet, on m’a demandé d’utiliser un serveur WEB local comme uWamp. Mais après mes discutions avec mon chef du projet et mon expert, j’utilisera **Render**. Avec cet utile je peux déployer mon site rapidement depuis mon dépôt GitHub et le site sera accessible dans quelque petite minute, mon site sera accessible sur internet. Mais pour les petits tests en local j’utiliserai l’extension VS code que j’ai déjà nommée.

Figure - Render.com

### figma kostenlos IconFigma

Figma c’est un outil récemment connu entre les développeurs WEB. Il permet de créer des maquets/interface pour les site WEB. Il est disponible sur Windows, MacOS et Linux pour créer des interfaces et sur Android/iOS permettant de visualiser des prototypes Figma.

Note : Figma a la capacité de tourner le prototype céé en code HTML et CSS. Mais je l’utiliserai uniquement pour créer le schéma de mon site web et savegrader en format image.

Figure - Figma

### Carbon

Pour avoir une meilleure présentation d’un extrait de mon code dans ce rapport, j’ai utilisé un site web qui s’appelle **Carbon** (carbon.now.sh). Je peux copier des morceaux de code et les coller dans Carbon, ensuite je peux télécharger une image claire et bien lisible.

Figure - Carbon.now.sh

### Préparation de matériel

* **Editeur du texte** : Les configurations pour mon éditeur du texte étaient assez simples car j’ai toujours la même configuration même si je travaille chez moi ou à l’école. Il me fallait donc connecter le Visual Studio Code à mon compte GitHub pour synchroniser les configurations et installer toutes les extensions que j’utilise constamment.
* **Git**: Mon dépôt git à une structure simple et je ne vais pas utiliser des différentes branches parce que ce n’est pas un projet trop compliqué à gérer.

## Préparation de l’environnement

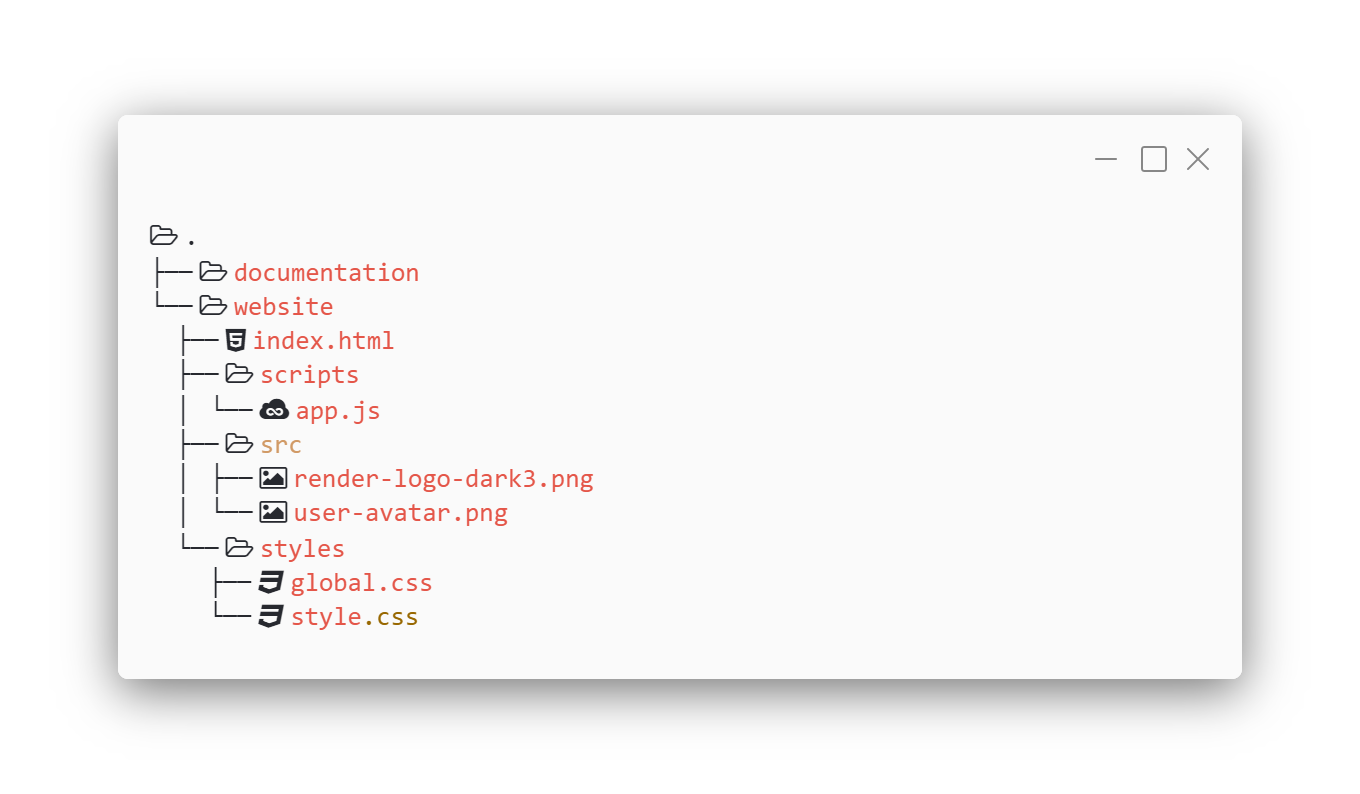


Figure - Répertoires du travail

### Documentation

### Développement

Pour mon environnement du travail, j’ai une structure assez simple. Dans le répertoire racine, j’ai deux dossiers, le premier dossier s’appelle « documentation » qui contient tous mes documents comme mon rapport, planification et journal du travail. Le deuxième dossier contient les fichiers de mon site. Dans ce dossier j’ai organisé les fichiers pour « CSS » dans le dossier « styles », les fichiers « JavaScript » dans le dossier « scripts » et les images dans le dossier « src ». Et en fin mon fichier index Html se trouve dans la racine de dossier « website ».

Le fichier « index.html » est la page que vous verrez quand vous entrez dans le site. Le « app.js » qui contient toutes les fonctionnalités de mon site. Le ficher « global.css » contiendra la présentation globale de mon site et le fichier « style.css » contiendra la présentation spécifique de chaque élément dans mon site.

TODO : autres environnements

# Réalisation

## Dossier de Réalisation

### Version des matériels/librairies utilisées

* Visual Studio Code v1.67.1
* FontAwesome Icon Library v6.1.1 and v4.7.0
* Live Server (Five Server) v0.1.4

### Construction la partie statique du site web

J’ai commencé le développement de mon projet par mettre en places les différents éléments visuels de mon site et séparer le site par des différentes sections.

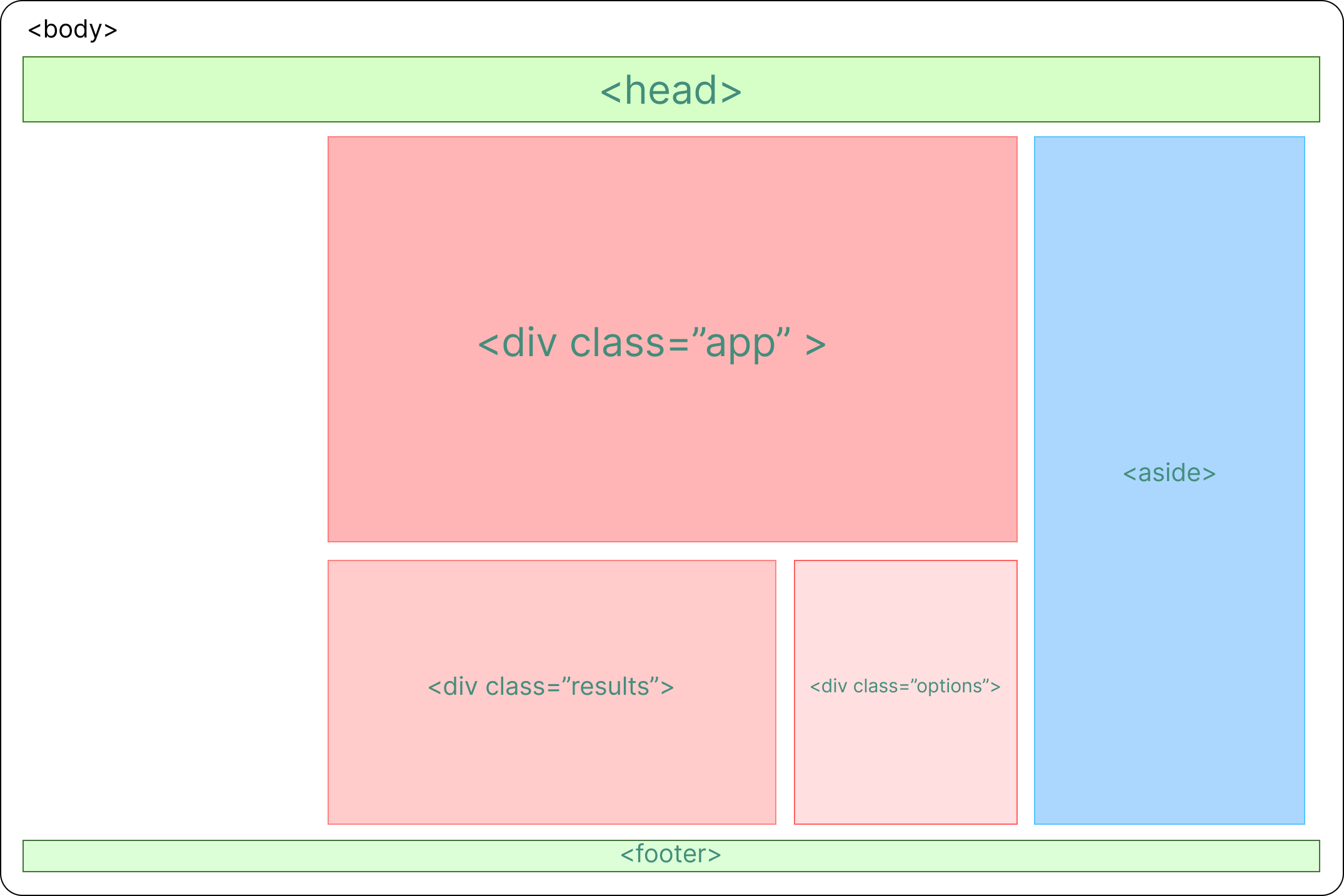


Figure - HTML<head>

Pour la balise ***head*** de mon fichier HTML :

1. J’ai ajouté deux liens vers les différentes versions de FontAwesome qui me permet s’accéder à une grande librairie des icônes pour mon site.
2. Deux liens pour accéder à mes fichiers css, « global.css » et « style.css ».
3. Un lien vers mon fichier JavaScript, « app.js ». Dans cette ligne j’ai aussi ajouté l’attribut « defer », cet attribut « lazy load » ma page JavaScript et permet au fichier HTML de charger complètement et ensuite charger le fichier JavaScript. La raison pour laquelle je charge le fichier script après le fichier html est la lourdeur de ma page script, cela peut créer des problèmes car mon fichier script ne verra pas les éléments html qui sont en dessous de lui et ne pourra donc pas les gérer.
4. Les balises ***META*** qui sont pour définir l’encodage de caractère dans le site. Ajouter la comptabilité avec des différentes versions de l’Internet Explorer. Comptabilité aves des appareils mobiles ou des appareils avec des différentes tailles de l’écran. L’auteur et description du site.
5. Lien pour afficher l’icône de mon site.

J’ai divisé la balise ***body*** en quatre sections principales, ces sections sont ***header***, ***main***, ***aside*** et ***footer***.



<body>

Figure - HTML<body> plan

La balise ***header*** contiendra logo, nome et la barre de navigation de mon site



Figure - HTML<body>header>

La balise ***<main>*** qui fait une grande partie du site, contient mon application donc le TypingTest. Cette balise s’est divisé en plusieurs <div> qui me permettront de bien les positionner dans différents endroits en utilisant CSS et pouvoir facilement accéder chaque élément depuis mon fichier Java Script.

Dans la balise ***main***, un ***div*** qui celui s’est divisé en différentes parties. Un header pour le message bienvenu de mon site web, un élément ***input*** dans laquelle les mots tapés par l’utilisateur seront capturés, un autre ***div*** pour visualisation de mon texte, un minuteur et en fin un bouton pour redémarrer mon test

Mon bouton a un ***onclick*** évent permettant à l’utilisateur d’exécuter une fonction JavaScript, dans mon cas, la fonction s’appelle ***newQuote()*** et je vais en parler dans la chapitre **4.1.4 JavaScript**.

******

Figure - HTML<body>main>div.app>

# Tests

## Dossier des tests

# Conclusion

## Bilan des fonctionnalités demandées

## Bilan de la planification

## Bilan personnel

# Divers

## Journal de travail

## Table des illustrations

[Figure 1 - Planification Initiale 5](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc104188134)

[Figure 2 - Planification détaillée (Semaine 1) 8](#_Toc104188135)

[Figure 3 - Planification détaillée (Semaine 2) 9](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc104188136)

[Figure 4 - Planification détaillée (Semaine 5) 10](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc104188137)

[Figure 5 - Planification détaillée (Semaine 4) 10](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc104188138)

[Figure 6 - Planification détaillée (Semaine 3) 10](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc104188139)

[Figure 7 - Visual Studio Code 11](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc104188140)

[Figure 8 – GitHub.com 11](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc104188141)

[Figure 9 - Render.com 11](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc104188142)

[Figure 10 - Figma 11](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc104188143)

[Figure 11 - Carbon.now.sh 11](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc104188144)

[Figure 12 - Répertoires du travail 13](#_Toc104188145)

[Figure 13 - HTML<head> 14](#_Toc104188146)

[Figure 14 - HTML<body> plan 15](#_Toc104188147)

[Figure 15 - HTML<body>header> 15](#_Toc104188148)

[Figure 16 - HTML<body>main>div.app> 16](#_Toc104188149)

## Glossaire

|  |  |
| --- | --- |
| MOTS |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Bibliographie

## Webographie

# Annexes