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](#_Toc103783395)

[1.1 Introduction 4](#_Toc103783396)

[1.2 Objectifs 4](#_Toc103783397)

[1.3 Prérequis 4](#_Toc103783398)

[2 Planification Initiale 4](#_Toc103783399)

[3 Analyse/Conception 6](#_Toc103783400)

[3.1 Concept 6](#_Toc103783401)

[3.1.1 Mockup 6](#_Toc103783402)

[3.2 Matériel et logiciels à disposition 7](#_Toc103783403)

[3.3 Risques techniques 7](#_Toc103783404)

[3.4 Stratégie de test 7](#_Toc103783405)

[3.5 Planification détaillée 8](#_Toc103783406)

[3.6 Dossier de conception 11](#_Toc103783407)

[3.6.1 Environnement de développement 11](#_Toc103783408)

[3.6.2 Utile de versioning 11](#_Toc103783409)

[3.6.3 Serveur WEB 11](#_Toc103783410)

[3.6.4 Préparation des matériels 11](#_Toc103783411)

[3.7 Préparation de l’environnement 12](#_Toc103783412)

[4 Réalisation 13](#_Toc103783413)

[4.1 Dossier de Réalisation 13](#_Toc103783414)

[4.1.1 HTML 13](#_Toc103783415)

[5 Tests 16](#_Toc103783416)

[5.1 Dossier des tests 16](#_Toc103783417)

[6 Conclusion 17](#_Toc103783418)

[6.1 Bilan des fonctionnalités demandées 17](#_Toc103783419)

[6.2 Bilan de la planification 17](#_Toc103783420)

[6.3 Bilan personnel 17](#_Toc103783421)

[7 Divers 17](#_Toc103783422)

[7.1 Journal de travail 17](#_Toc103783423)

[7.2 Table des illustrations 17](#_Toc103783424)

[7.3 Glossaire 17](#_Toc103783425)

[7.4 Bibliographie 18](#_Toc103783426)

[7.5 Webographie 18](#_Toc103783427)

[8 Annexes 19](#_Toc103783428)

# 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 du test : par nombres de mots, par compte à rebours
2. Un texte à recopier : les mots français et les mot générés avec des caractères aléatoires
3. Un chronomètre qui se démarre quand je commence à taper et s’arrête quand je fini de taper ou le temps arrive à zéro dépanadant quelle mode j’ai choisie
4. Gestion des utilisateurs à l’aide des cookies

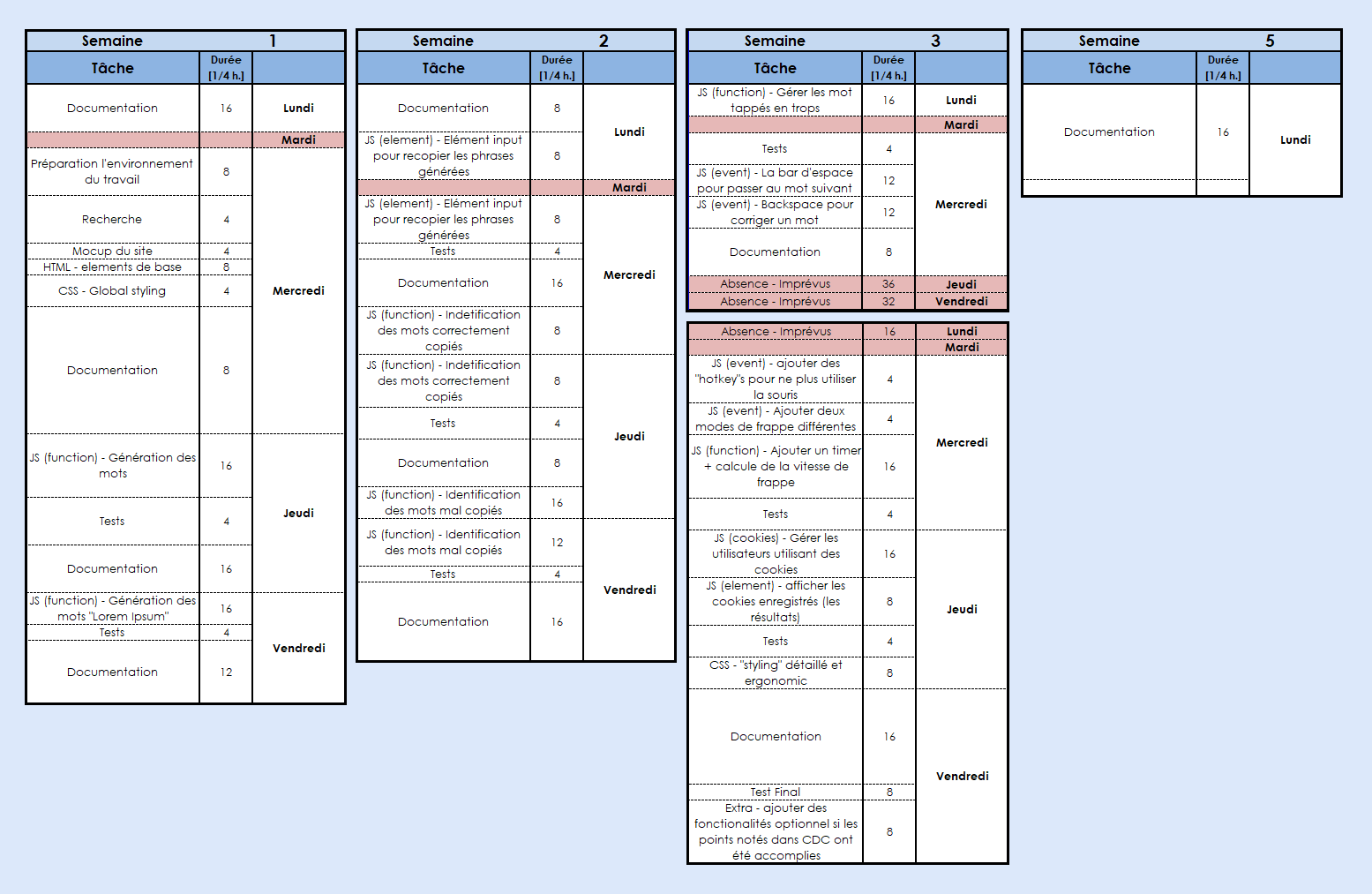
## Prérequis

* 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 ceux que j’ai globalement prévus comme tâches et ça me permettra de voir une liste de mes tâches à faire durant ce projet.



**Semaine 4**

Figure 1 - Planification Initiale

**Semaine 4**

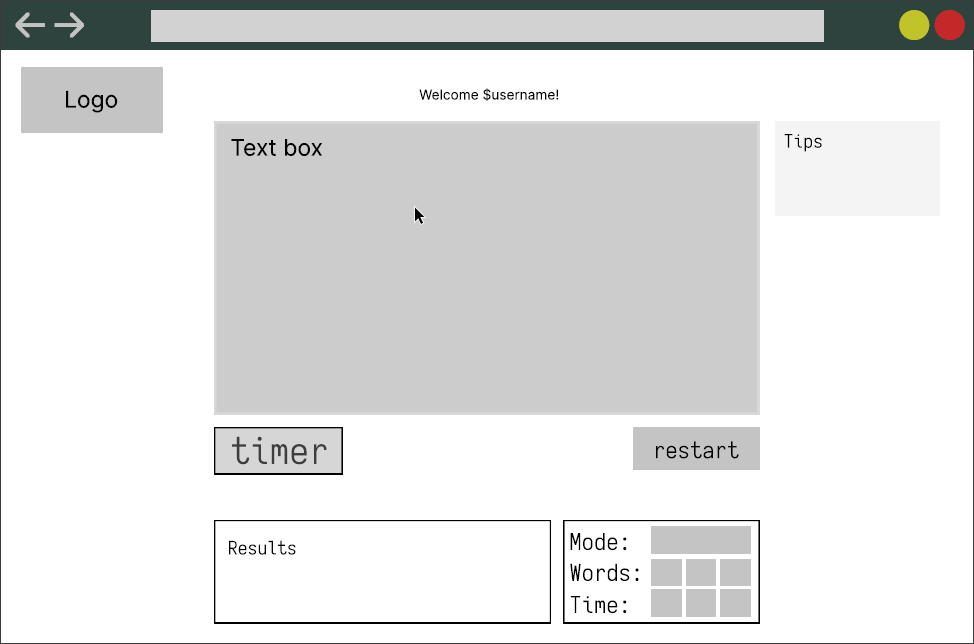
# Analyse/Conception

## Concept

Plan du site est relativement simple et tout se passe dans une page.

### Mockup





## 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

## Risques techniques

Pendant 4-5 ans de ma présence à l’ETML, la 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 me passionnait toujours, même si je n’étais pas assez bien dans le domaine. Donc pendant mon stage d’entreprise j’ai pu beaucoup travailler sur JavaScript et m’en 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 3 - Planification détaillée (Semaine 2)



Figure - Planification détaillée (Semaine 4)

Figure - Planification détaillée (Semaine 5)

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.

### Environnement de développement

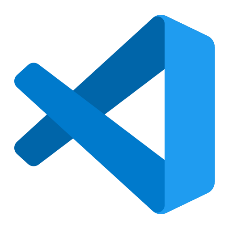
Personnellement j’utilise souvent NeoVIM qui est une continuation d’un utile de traitement du texte connue s’appelé VIM. Main pour ce projet j’ai décidé de continuer avec la proposition dans mon cahier de charge. **Visual Studio Code**, car NeoVIM n’est pas un utile optimisé pour être utilisé dans environnement 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 7 - Visual Studio Code

### Outil 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 8 – 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 9 - Render.com

### Préparation de matériel

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.

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

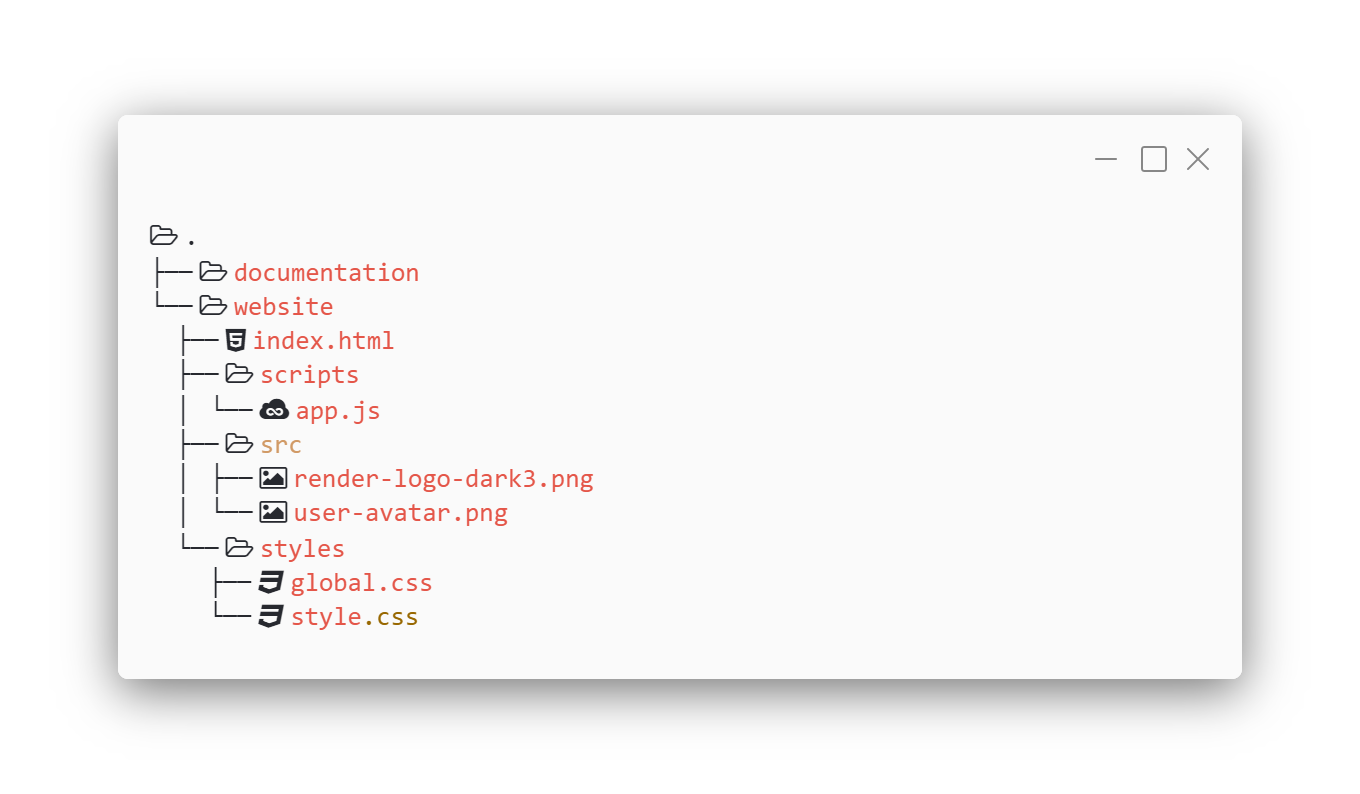


Figure - Répertoires du travail

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

### HTML

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.

#### Head



Figure - HTML<head>

Pour le « 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.

#### Body

J’ai divisé le « body » en quatre sections principales, ces sections sont « header ». « Main », « aside » et « footer ».

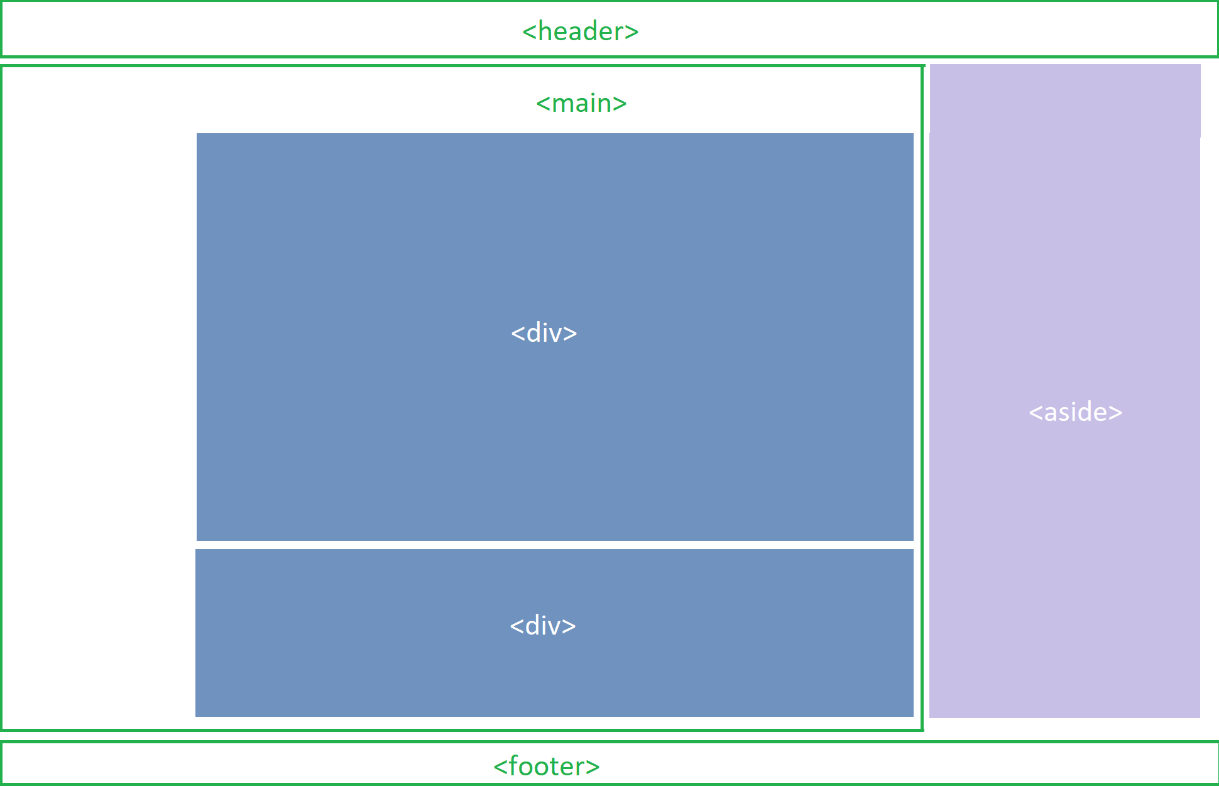


Figure - HTML<body> plan

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.

* 1. ***<div id= ‘’app’’>*** contient un header pour le message de bienvenue. Un <input> de texte. Un <div id=’’quote’’> avec une balise paragraphe (<p>) pour afficher le texte à recopier.

******

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#_Toc103782912)

[Figure 2 - Planification détaillée (Semaine 1) 8](#_Toc103782913)

[Figure 3 - Planification détaillée (Semaine 2) 9](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc103782914)

[Figure 4 - Planification détaillée (Semaine 4) 10](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc103782915)

[Figure 5 - Planification détaillée (Semaine 5) 10](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc103782916)

[Figure 6 - Planification détaillée (Semaine 3) 10](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc103782917)

[Figure 7 - Visual Studio Code 11](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc103782918)

[Figure 8 – GitHub.com 11](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc103782919)

[Figure 9 - Render.com 11](file:///C:\Users\nooalizadeh\OneDrive%20-%20Education%20Vaud\2021-2022\TPI\TypingTest\documentation\nooalizadeh-rapport.docx#_Toc103782920)

[Figure 10 - Répertoires du travail 12](#_Toc103782921)

[Figure 11 - HTML<head> 13](#_Toc103782922)

[Figure 12 - HTML<body> plan 14](#_Toc103782923)

[Figure 13 - HTML<body>header> 14](#_Toc103782924)

[Figure 14 - HTML<body>main>div.app> 15](#_Toc103782925)

## Glossaire

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

## Bibliographie

## Webographie

# Annexes