

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
1.1	INTRODUCTION	4
1.2	OBJECTIFS	4
1.3	PRÉREQUIS.....	4
2	PLANIFICATION INITIALE	4
3	ANALYSE/CONCEPTION	6
3.1	CONCEPT	6
3.1.1	<i>Mockup</i>	6
3.2	MATÉRIEL ET LOGICIELS À DISPOSITION	7
3.3	RISQUES TECHNIQUES.....	7
3.4	STRATÉGIE DE TEST	7
3.5	PLANIFICATION DÉTAILLÉE	8
3.6	DOSSIER DE CONCEPTION.....	11
3.6.1	<i>Environnement de développement</i>	11
3.6.2	<i>Utile de versioning</i>	11
3.6.3	<i>Serveur WEB</i>	11
3.6.4	<i>Préparation des matériels</i>	11
3.7	PRÉPARATION DE L'ENVIRONNEMENT.....	12
4	RÉALISATION.....	13
4.1	DOSSIER DE RÉALISATION	13
4.1.1	<i>HTML</i>	13
5	TESTS.....	16
5.1	DOSSIER DES TESTS	16
6	CONCLUSION	17
6.1	BILAN DES FONCTIONNALITÉS DEMANDÉES	17
6.2	BILAN DE LA PLANIFICATION	17
6.3	BILAN PERSONNEL.....	17
7	DIVERS	17
7.1	JOURNAL DE TRAVAIL.....	17
7.2	TABLE DES ILLUSTRATIONS	17

7.3	GLOSSAIRE	17
7.4	BIBLIOGRAPHIE.....	18
7.5	WEBOGRAPHIE.....	18
8	ANNEXES.....	19

1 ANALYSE PRÉLIMINAIRE

1.1 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.

1.2 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

1.3 PRÉREQUIS

- Modules ICT orienté WEB
- Usage des Cookies
- JavaScript

2 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.

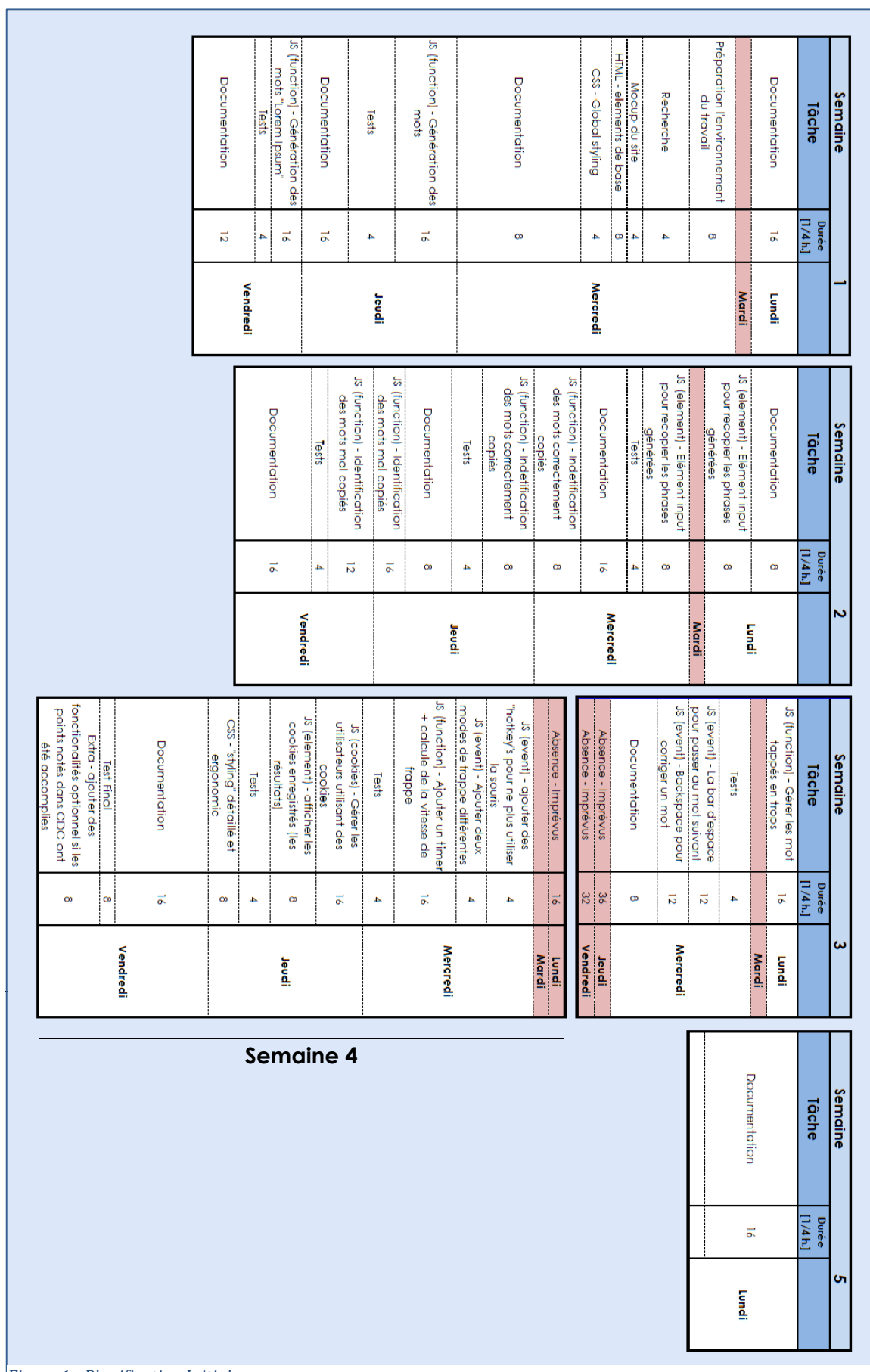


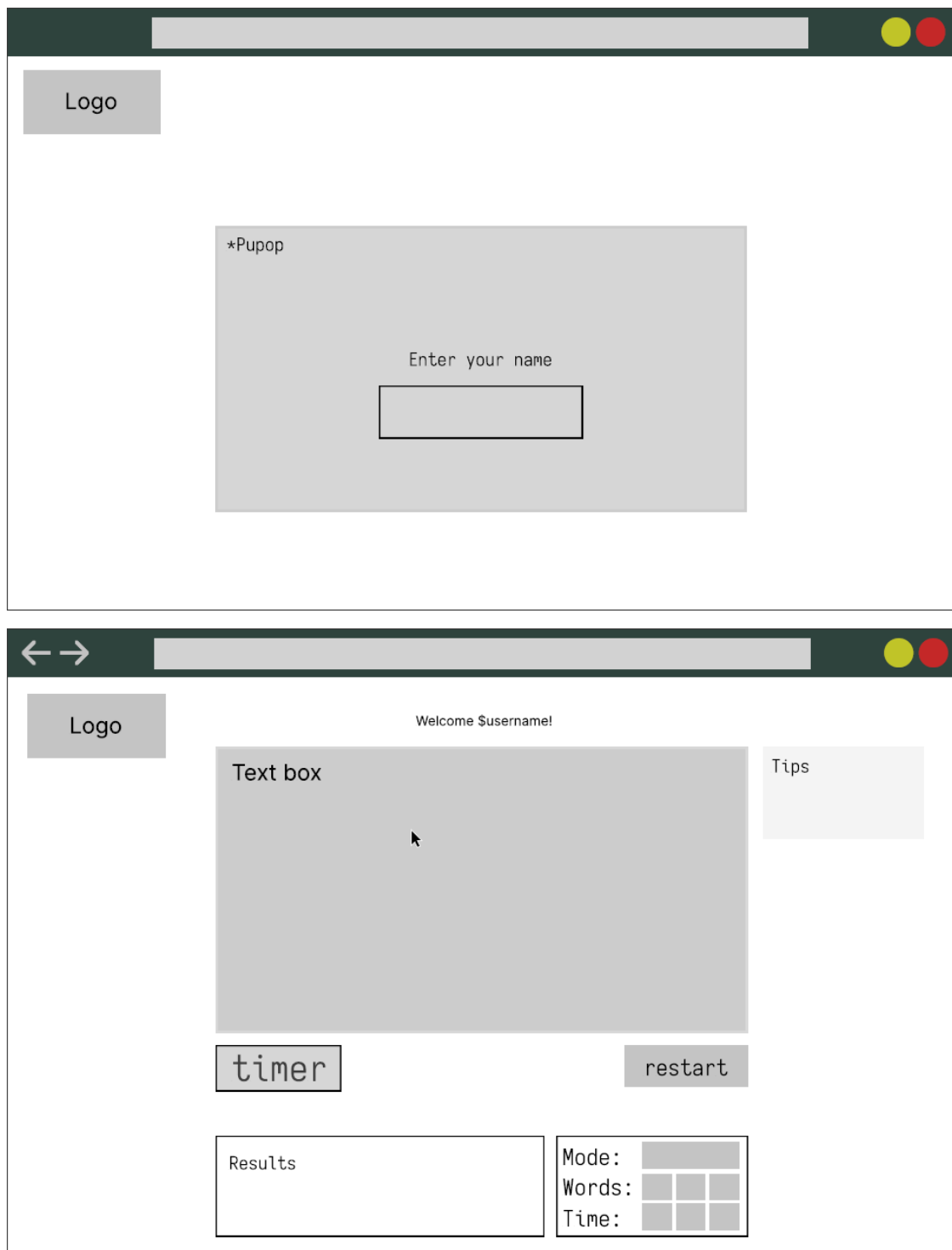
Figure 1 - Planification Initiale

3 ANALYSE/CONCEPTION

3.1 CONCEPT

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

3.1.1 Mockup



3.2 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

3.3 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.

3.4 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.

3.5 PLANIFICATION DÉTAILLÉE

Semaine		1	Explications: qu'est-ce qui se fait et comment ?
Tâche	Durée [1/4 h.]		
Documentation	16	Lundi	Matin: Visite avec l'expert Fin matinée: Commencer à faire la planification initiale
Préparation l'environnement du travail	8		- Préparation du git et mes répertoires du travail - Configuration ssh pour accéder au git depuis terminal en local - Installation et configuration Vscode
Recherche	4		- Recherche des sources et des images différents pour le site - Trouver un utilité pour créer le mockup du site - Chercher des idées pour le look du site
Mocup du site	4		
HTML - éléments de base	8		- Commencer à créer la page HTML et mettre les balises nécessaires
CSS - Global styling	4		- Ajouter un styling global pour tout le site et pouvoir distinguer les éléments html sur la page
Documentation	8	Mercredi	- Commencer à écrire le rapport du projet - Spécification - CDC - Analyse - Noter les tâches que j'ai fait pour Mercredi - Réalisation - enregistrer la dernière version de ces doc en format PDF pour les envoyer au chef du projet et aux experts
JS (function) - Génération des mots	16		- Créer la fonction quoteGen() qui s'active en appuyant sur un bouton et prendre les citations depuis un fichier json et les affiche aléatoirement dans le site. - Créer la fonction wordGen() qui permet de générer les mots français aléatoires et les affiche dans le site web
Tests	4	Jeudi	- Tester voir si les citations sont correctement afficher et à chaque rafraichissement du site une nouvelle citation s'affiche - Le même test mais pour des mots français - Noter les tâches dans le journal du travail - Avancer dans le rapport et noter ces dernières tâches en plus de la résultat de leurs tests - Tests
Documentation	16		- Créer la fonction charGen() qui génère des mots apartir des différents types de characters
JS (function) - Génération des mots "Lorem Ipsum"	16		- Tester si à chaque rafraichissement une nouvelle phrase se génère
Tests	4		- Mettre à jour le journal du travail
Documentation	12	Vendredi	- Rapport - Réalisation - Préparer les PDF pour les envoyer
Total semaine	120		Max. 120

Figure 2 - Planification détaillée (Semaine 1)

Semaine		2	
Tâche	Durée [1/4 h.]		Explications: qu'est-ce qui se fait et comment ?
Documentation	8	Lundi	_Mettre à jour le rapport > Tests > Sources
JS (element) - Elément input pour recopier les phrases générées	8		_Ajouter un moyen de taper le texte dans une case et pouvoir le comparer avec les textes générés
JS (element) - Elément input pour recopier les phrases générées	8	Mercredi	_** continuation la tâche précédente
Tests	4		_donner un message d'alert quand le texte tapé n'est pas égale à la phrase générée
Documentation	16		_Mettre à jour le journal du travail avec les tâches du lundi et mercredi _Noter les résultats de testes que j'ai fait pour cette tâche _Préparer les DPF
JS (function) - Indetification des mots correctement copiés	8		_Ajouter un moyen d'identifier les character correctement tapés _**
JS (function) - Indetification des mots correctement copiés	8	Jeudi	_Tester la distinction entre les mot corrects et les autres, couleur jaune pour les charachers correctement copié
Tests	4		_Rapport > Réalisation > Tests
Documentation	8		_Ajouter un moyen d'identifier les character mal tapés
JS (function) - Identification des mots mal copiés	16		_Continuer sur l'identification des mots mal copiés
JS (function) - Identification des mots mal copiés	12	Vendredi	_Distinguer les mots correctement tapés avec des mot incorrecte, les mots mal tapé seront barés
Tests	4		_Tester si on voit bien la difference des mots correcte et incorrectes
Documentation	16		_Noter les nouvelles tâches _Rapport > Réalisation > Tests
Documentation	16		_Préparation PDF
Total semaine	120		Max. 120

Figure 3 - Planification détaillée (Semaine 2)

Semaine 3			
Tâche	Durée [1/4 h.]		Explications: qu'est-ce qui se fait et comment ?
JS (function) - Gérer les mot tapés en trop	16	Lundi	Créer la fonction qui detecte les characters tapés en trop et pousse les mots suivants si je continue à taper des characters en trop
Tests	4	Mercredi	_Tester les mots sont bien poussés, les characters en trop devraient pas aller en dessus des mots suivants
JS (event) - La bar d'espace pour passer au mot suivant	12		_Quand un mot est correctement tapé, en tapant sur l'espace je passe au mot suivant et je continue à taper ce nouveau mot
JS (event) - Backspace pour corriger un mot	12		_En tapant sur backspace, je pourrais corriger un mot
Documentation	8		_Journal des tâches effectués cette semaine _Noter les trois fonctionnalités ajouter cette semaine > Réalisation
Absence - Imprévus	36	Jeudi	Jeudi de l'Ascension
Absence - Imprévus	32	Vendredi	Pont de l'Ascension
Total semaine	120		Max. 120

Figure 6 - Planification détaillée (Semaine 3)

Semaine 4			
Tâche	Durée [1/4 h.]		Explications: qu'est-ce qui se fait et comment ?
Absence - Imprévus	16	Lundi	Lundi de Pentecôte
JS (event) - ajouter des "hotkey"s pour ne plus utiliser la souris	4	Mercredi	_Créer des "event handlers" pour naviguer dans le site sans utiliser la souris: charger nouvelle phrase à taper, refaire un teste de frappe, arreter le timer, etc...
JS (event) - Ajouter deux modes de frappe différentes	4		_Ajouter la fonctionnalité de passer aux différentes modes de frapper différents : normale, mots français. Lorem Ipsum
JS (function) - Ajouter un timer + calcule de la vitesse de frappe	16		_Ajouter un timer qui commence à compter dès que la personne commence à taper un mot et il s'arrête quand il fini à taper la phrase en entier ou il réussi pas à taper les mot avant que le timer arrive à zéro.
Tests	4		_Calculer nombre des mots tapés par minutes et donner le MPM, vitesse de frappe
JS (cookies) - Gérer les utilisateurs utilisant des cookies	16	Jeudi	_Tester que le timer est correcte, il arrête bien au moment dont le dernier character est tapé
JS (element) - afficher les cookies enregistrés (les résultats)	8		_Demander aux utilisateurs de taper leur nom pour accéder au site et ça sera enregistré dans un cookie
Tests	4		_Pouvoir afficher tout les cookies enregistrés dans le site dans la page de resultats
CSS - "styling" détaillé et ergonomic	8		_tester les cookies sont biens associé à la bonne personne ainsi ses resultats MPM, PDF(precision de frappe), DEH(date et heure)
Documentation	16	Vendredi	_Commencer à travailler sur l'ergonomie du site et le rendre agréable aux yeux
Test Final	8		_Journal du travail _Rapport > Réalisation > Tests > Conclusion > Blians
Extra - ajouter des fonctionnalités optionnel si les points notés dans CDC ont été accomplies	8		_Faire les tests finale pour donner un vue ensemble des testes dans le rapport _Si le temps me permet, j'ajouterai certains fonctionnalité dans le site
Total semaine	112		Max. 120

Figure 4 - Planification détaillée (Semaine 4)

Semaine 5			
Tâche	Durée [1/4 h.]		Explications: qu'est-ce qui se fait et comment ?
Documentation	16	Lundi	_Remplir et finir le journal du travail _Faire des dernières modifications dans le rapport, ajouter les photos de mes tâches (JDT), finir la partie des tests _Mise en page du rapport _Préparer les PDFs pour les envoyer aux experts et mon chef du projet _Imprimer mes documents pour les envoyer par courrier aux experts

Figure 5 - Planification détaillée (Semaine 5)

3.6 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.

3.6.1 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.

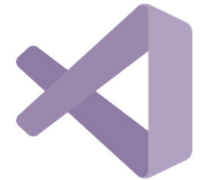


Figure 7 - Visual Studio Code

3.6.2 Utile 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.

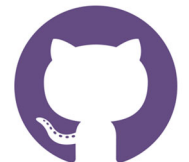


Figure 8 – GitHub.com

3.6.3 Serveur WEB

Dans ce projet, on m'a demandé d'utiliser un serveur WEB local comme uWamp. Mais après mes discussions 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

3.6.4 Préparation des matériels

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.

TODO

3.7 PRÉPARATION DE L'ENVIRONNEMENT

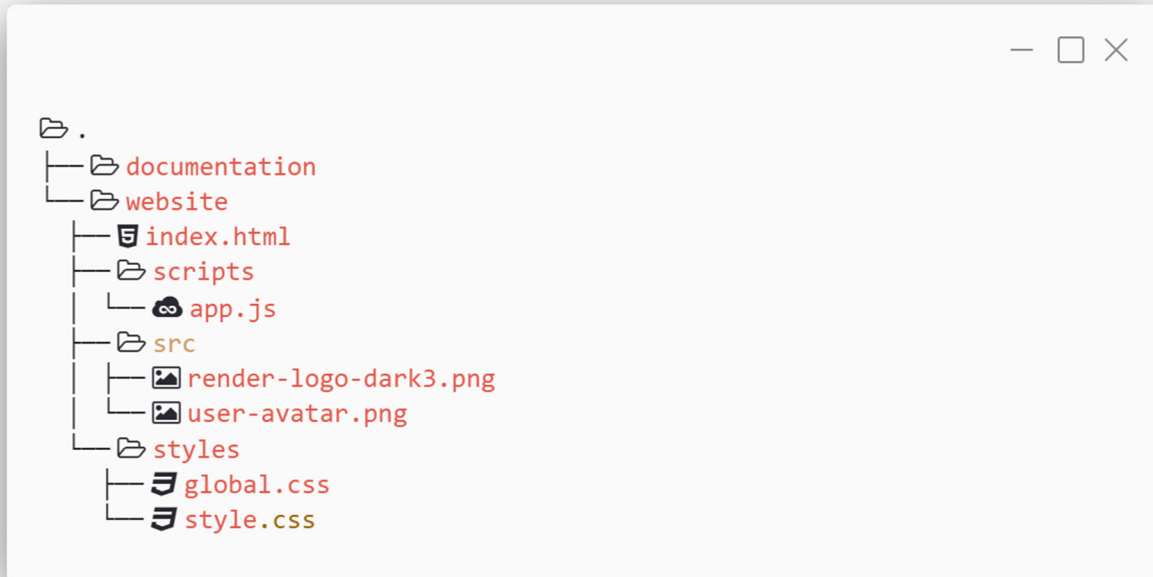


Figure 10 - Répertoires du travail

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 fichier « 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

4 RÉALISATION

4.1 DOSSIER DE RÉALISATION

4.1.1 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.

4.1.1.1 Head



```
<head>
  <link rel="stylesheet" href="//use.fontawesome.com/releases/v4.7.0/css/all.css">
  <link rel="stylesheet" href="//use.fontawesome.com/releases/v5.15.4/css/all.css">
  <link rel="stylesheet" href="/styles/global.css">
  <link rel="stylesheet" href="/styles/style.css">
  <script src="/scripts/app.js" defer></script>
  <title>TypingTest</title>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="author" content="NoorMohammad Alizadeh">
  <meta name="description" content="Un site pour tester la vitesse de frappe.">
  <link rel="shortcut icon" href="/src/favicon.png" type="image/x-icon">
</head>
```

Figure 11 - HTML<head>

Pour le « head » de mon fichier HTML :

- 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.
- Deux liens pour accéder à mes fichiers css, « global.css » et « style.css ».
- 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.
- 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 avec des appareils mobiles ou des appareils avec des différentes tailles de l'écran. L'auteur et description du site.
- Lien pour afficher l'icône de mon site.

4.1.1.2 Body

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

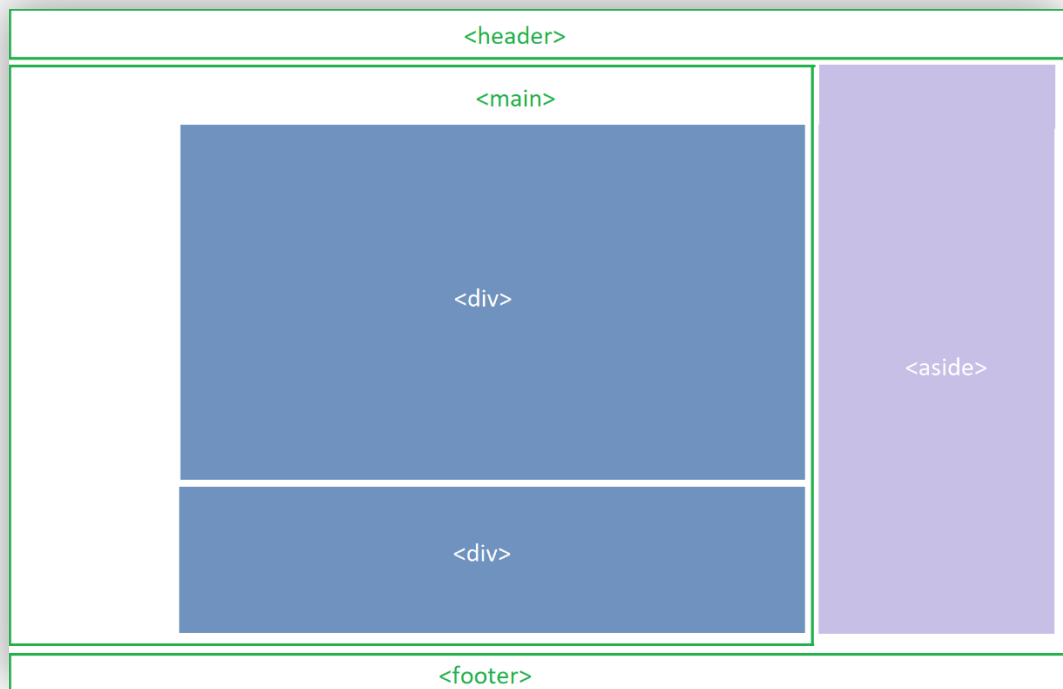


Figure 12 - HTML<body> plan

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

```
<header>
  
  <p>TypingTest</p>
  <ul>
    <li><a href="#" id="aboutPop" onclick="aboutPop()">À propos</a></li>
    <li><a href="https://etml.ch">ETML</a></li>
  </ul>
</header>
```

Figure 13 - 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.

- a. **<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.

```
<div class="app">
  <div class="header">
    <h3 id="greetMessage">Hello <span id="username">username</span></h3>
  </div>
  <input type="text" id="inputQuote" name="InputQuote">
  <div id="quote">
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Voluptatibus non commodi,
    dolores porro architecto id! Laboriosam consequatur, numquam unde magnam perspiciatis temporibus et
    corporis aperiam, id possimus adipisci repellat! Minima.</p>
  </div>
  <p id="timer">00</p>
  <button name="Restart" id="restartBtn" onclick="newQuote()">Redémarrer</button>
</div>
```

Figure 14 - HTML<body>main>div.app>

5 TESTS

5.1 DOSSIER DES TESTS

6 CONCLUSION

6.1 BILAN DES FONCTIONNALITÉS DEMANDÉES

6.2 BILAN DE LA PLANIFICATION

6.3 BILAN PERSONNEL

7 DIVERS

7.1 JOURNAL DE TRAVAIL

7.2 TABLE DES ILLUSTRATIONS

FIGURE 1 - PLANIFICATION INITIALE	5
FIGURE 2 - PLANIFICATION DÉTAILLÉE (SEMAINE 1)	8
FIGURE 3 - PLANIFICATION DÉTAILLÉE (SEMAINE 2)	9
FIGURE 4 - PLANIFICATION DÉTAILLÉE (SEMAINE 4)	10
FIGURE 5 - PLANIFICATION DÉTAILLÉE (SEMAINE 5)	10
FIGURE 6 - PLANIFICATION DÉTAILLÉE (SEMAINE 3)	10
FIGURE 7 - VISUAL STUDIO CODE	11
FIGURE 8 – GITHUB.COM	11
FIGURE 9 - RENDER.COM	11
FIGURE 10 - RÉPERTOIRES DU TRAVAIL	12
FIGURE 11 - HTML<HEAD>	13
FIGURE 12 - HTML<BODY> PLAN	14
FIGURE 13 - HTML<BODY>HEADER>	14
FIGURE 14 - HTML<BODY>MAIN>DIV.APP>	15

7.3 GLOSSAIRE

MOTS	

7.4 BIBLIOGRAPHIE

7.5 WEBOGRAPHIE

8 ANNEXES