

# 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

<b>1</b>	<b>ANALYSE PRÉLIMINAIRE.....</b>	<b>3</b>
1.1	INTRODUCTION .....	3
1.2	OBJECTIFS .....	3
1.3	PRÉREQUIS.....	3
<b>2</b>	<b>PLANIFICATION INITIALE .....</b>	<b>3</b>
<b>3</b>	<b>ANALYSE/CONCEPTION .....</b>	<b>5</b>
3.1	CONCEPT .....	5
3.1.1	<i>Mockup</i> .....	5
3.2	MATÉRIEL ET LOGICIELS À DISPOSITION.....	6
3.3	RISQUES TECHNIQUES .....	6
3.4	STRATÉGIE DE TEST .....	6
3.5	PLANIFICATION DÉTAILLÉE .....	7
3.6	DOSSIER DE CONCEPTION .....	10
3.6.1	<i>Environnement de développement</i> .....	10
3.6.2	<i>Utile de versioning</i> .....	10
<b>4</b>	<b>RÉALISATION .....</b>	<b>11</b>
4.1	DOSSIER DE RÉALISATION .....	11
<b>5</b>	<b>TESTS.....</b>	<b>12</b>
5.1	DOSSIER DES TESTS .....	12
<b>6</b>	<b>CONCLUSION .....</b>	<b>13</b>
6.1	BILAN DES FONCTIONNALITÉS DEMANDÉES .....	13
6.2	BILAN DE LA PLANIFICATION .....	13
6.3	BILAN PERSONNEL.....	13
<b>7</b>	<b>DIVERS .....</b>	<b>13</b>
7.1	JOURNAL DE TRAVAIL.....	13
7.2	TABLE DES ILLUSTRATIONS .....	13
7.3	GLOSSAIRE .....	13
7.4	BIBLIOGRAPHIE.....	13
7.5	WEBOGRAPHIE.....	14
<b>8</b>	<b>ANNEXES.....</b>	<b>15</b>

## 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.

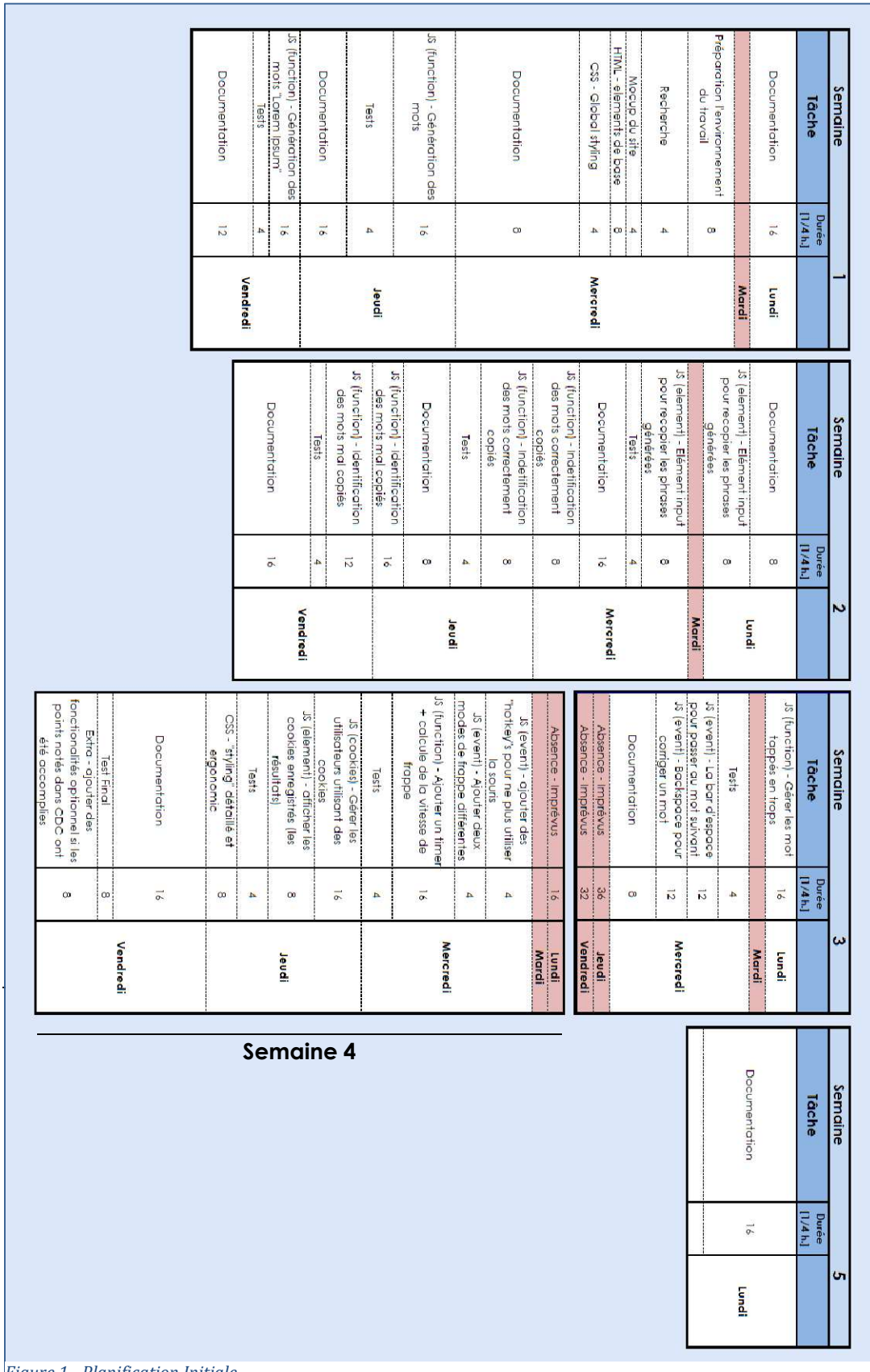


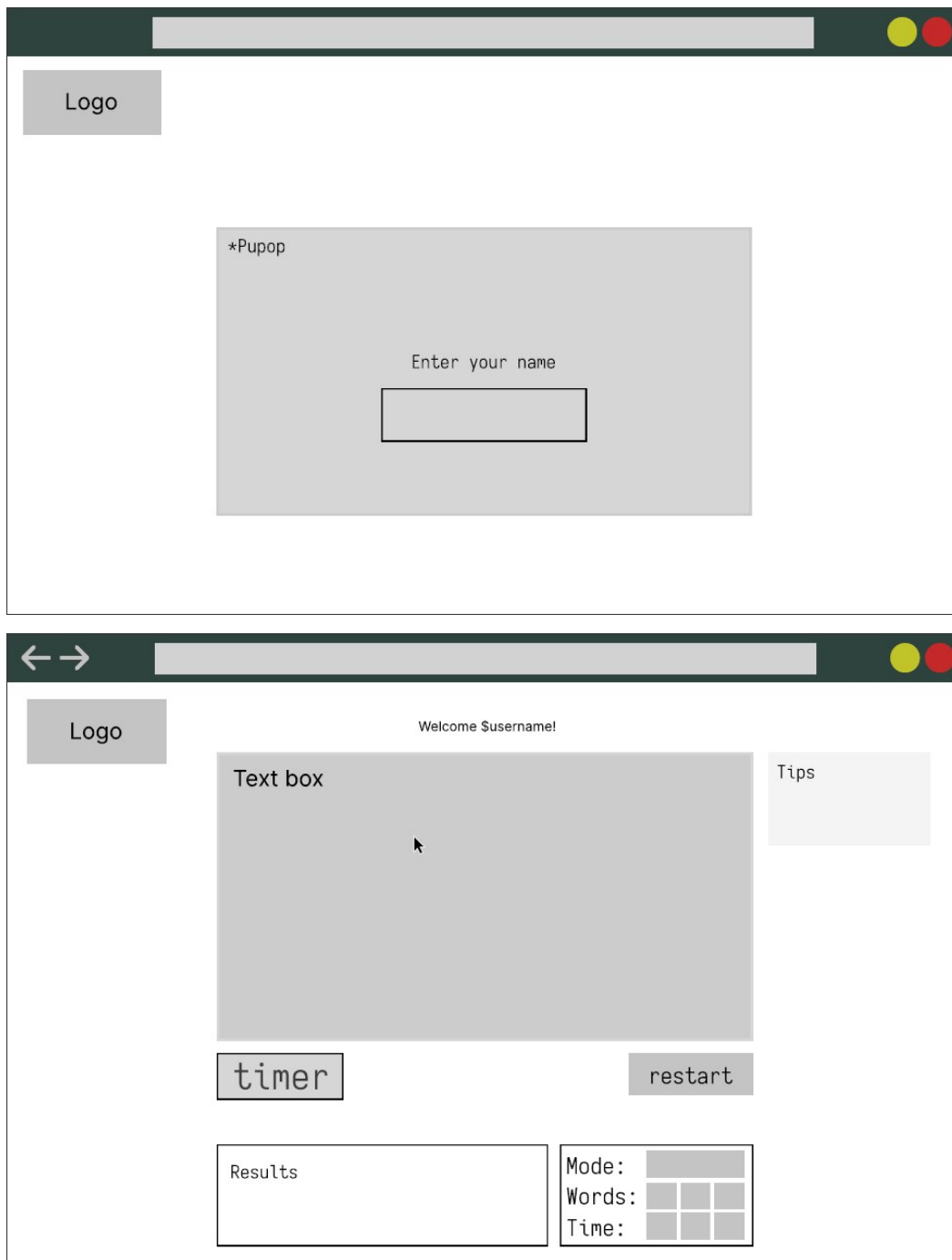
Figure 1 - Planification Initiale

## 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	<b>Lundi</b>	Matin: Visite avec l'expert Fin matinée: Commencer à faire la planification initiale
Préparation l'environnement du travail	8	<b>Mercredi</b>	<ul style="list-style-type: none"> <li>Préparation du git et mes répertoires du travail</li> <li>Configuration ssh pour accéder au git depuis terminal en local</li> <li>Installation et configuration Vscode</li> </ul>
Recherche	4		<ul style="list-style-type: none"> <li>Recherche des sources et des images différents pour le site</li> <li>Trouver un util pour créer le mockup du site</li> <li>Chercher des idées pour le look du site</li> </ul>
Mocup du site	4		
HTML - elements de base	8		<ul style="list-style-type: none"> <li>Commencer à créer la page HTML et mettre les balises nécessaires</li> </ul>
CSS - Global styling	4		<ul style="list-style-type: none"> <li>Ajouter un styling global pour tout le site et pouvoir distinguer les elements html sur la page</li> </ul>
Documentation	8	<b>Jendredi</b>	<ul style="list-style-type: none"> <li>Commencer à écrire le rapport du projet</li> <li>&gt; Spécification</li> <li>&gt; CDC</li> <li>&gt; Analyse</li> <li>Noter les tâches que j'ai fait pour Mercredi</li> <li>&gt; Réalisation</li> </ul>
JS (function) - Génération des mots	16		<ul style="list-style-type: none"> <li>enregistrer la dernière version de ces doc en format PDF pour les envoyer au chef du projet et aux experts</li> <li>Créer la fonction quoteGen() qui s'active en appuyant sur un bouton et prendre les citations depuis un fichier json et les affiches aléatoirement dans le site.</li> <li>Créer la fonction wordGen() qui permet de gerere les mot français aléatoires et les affiche dans le site web</li> </ul>
Tests	4		<ul style="list-style-type: none"> <li>Tester voir si les citations sont correctement afficher et à chaque rafraichissement du site une nouvelle citation s'affiche</li> </ul>
Documentation	16		<ul style="list-style-type: none"> <li>Le même test mais pour des mots français</li> <li>Noter les tâches dans le journal du travail</li> <li>Avancer dans le rapport et noter ces dernières tâches en plus de la résultat de leurs tests</li> <li>&gt; Tests</li> </ul>
JS (function) - Génération des mots "Lorem Ipsum"	16	<b>Vendredi</b>	<ul style="list-style-type: none"> <li>Créer la fonction charGen() qui génère des mots apartir des différents types de characters</li> </ul>
Tests	4		<ul style="list-style-type: none"> <li>Tester si à chaque rafraichissement une nouvelle phrase se génère</li> </ul>
Documentation	12		<ul style="list-style-type: none"> <li>Mettre à jour le journal du travail</li> <li>Rapport</li> <li>&gt; Réalisation</li> <li>Préparer les PDF pour les envoyer</li> </ul>
Total semaine	120		<b>Max. 120</b>

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		_**_continuation la tâche précédente
Tests	4	Mercredi	_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 tests que j'ai fait pour cette tâche _Préparer les DPF
JS (function) - Indentification des mots correctement copiés	8		_Ajouter un moyen d'identifier les caractères correctement tapés
JS (function) - Indentification des mots correctement copiés	8		_**
Tests	4		_Testing la distinction entre les mots corrects et les autres, couleur jaune pour les caractères correctement copiés
Documentation	8	Jeudi	_Rapport > Réalisation > Tests
JS (function) - Indentification des mots mal copiés	16		_Ajouter un moyen d'identifier les caractères mal tapés
JS (function) - Indentification des mots mal copiés	12		_Continuer sur l'identification des mots mal copiés _Distinguer les mots correctement tapés avec des mots incorrects, les mots mal tapés seront barrés
Tests	4	Vendredi	_Testing si on voit bien la différence des mots corrects et incorrects _Noter les nouvelles tâches
Documentation	16		_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 tappant 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 tappant sur backspace, je pourrais corriger un mot
Documentation	8		Journal des tâches effectués cette semaine _ Noter les trois fonctionnalités ajouter cette semaine > Réalisation
Absence - Imprévis	36	Jeudi	Jeudi de l'Ascension
Absence - Imprévis	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 - Imprévis	16	Lundi	Lundi de Pentecôte
JS (event) - ajouter des "hotkey"s pour ne plus utiliser la souris	4	Mercredi	_ Creer 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. _ Calculer nombre des mots tapés par minutes et donner le MPM, vitesse de frappe
Tests	4		_ Tester que le timer est correcte, il arrête bien au moment dont le dernier character est tapé
JS (cookies) - Gérer les utilisateurs utilisant des cookies	16	Jeudi	_ Demander aux utilisateurs de taper leur nom pour accéder au site et ça sera enregistré dans un cookie
JS (element) - afficher les cookies enregistrés (les résultats)	8		_ Povoir afficher tout les cookies enregistrés dans le site dans la page de resultats
Tests	4		_ tester les cookies sont biens associé à la bonne personne ainsi ses resultats MPM, PDF(precision de frappe), DEH(date et heure)
CSS - "styling" détaillé et ergonomic	8		_ Commencer à travailler sur l'ergonomie du site et le rendre agréable aux yeux
Documentation	16	Vendredi	Journal du travail _ Rapport > Réalisation > Tests > Conclusion > Blians
Test Final	8		_ Faire les tests finale pour donner un vue ensemble des testes dans le rapport
Extra - ajouter des fonctionnalités optionnel si les points notés dans CDC ont été accomplies	8		_ 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.

### 3.6.2 Utile de versioning

Pour

## **4 RÉALISATION**

---

### **4.1 DOSSIER DE RÉALISATION**

## 5 TESTS

---

### 5.1 DOSSIER DES TESTS



## 7.5 WEBOGRAPHIE

## 8 ANNEXES

---