# INFORMATIONS GENERALES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Candidat** | Nom : |  | Prénom : |  |
|  | 📪 | Maggioli | 🕿 | Luca |
| **Lieu de travail :** | Kudelski Group Rte de Genève 22-24, 1033 Cheseaux sur Lausanne | | | |
| **Orientation :** | 88601 Développement d’applications  88602 Informatique d’entreprise  88603 Technique des systèmes | | | |
| **Chef de projet** | Nom : |  | Prénom : |  |
| 📪 | | 🕿 | |
| **Expert 1** | Nom : |  | Prénom : |  |
| 📪 | | 🕿 | |
| **Expert 2** | Nom : |  | Prénom : |  |
| 📪 | | 🕿 | |
| **Période de réalisation :** | Du mercredi 31 août 2022 au lundi 19 août 17h 2022 | | | |
| **Horaire de travail :** | Du lundi au jeudi de 8h00 à 12h00 et de 13h00 à 17h35.  Vendredi Jour de cour EPSIC.  Pause de 20 minutes le matin et pause de 15 minutes l’après-midi. | | | |
| **Nombre d'heures :** | 90 heures | | | |
| **Planning (en H ou %)** | Analyse : 10% | | | |
| Implémentation : 40% | | | |
| Tests : 20% | | | |
| Documentations : 30% | | | |

# PROCÉDURE

* Le candidat réalise un travail personnel sur la base d'un cahier des charges reçu le 1er jour.
* Le cahier des charges est approuvé par les deux experts. Il est en outre présenté, commenté et discuté avec le candidat. Par sa signature, le candidat accepte le travail proposé.
* Le candidat a connaissance de la feuille d’évaluation avant de débuter le travail.
* Le candidat est entièrement responsable de la sécurité de ses données.
* En cas de problèmes graves, le candidat avertit au plus vite les deux experts et son CdP.
* Le candidat a la possibilité d’obtenir de l’aide, mais doit le mentionner dans son dossier.
* A la fin du délai imparti pour la réalisation du TPI, le candidat doit transmettre par courrier électronique le dossier de projet aux deux experts et au chef de projet. En parallèle, une copie papier du rapport doit être fournie sans délai en trois exemplaires (L’un des deux experts peut demander à ne recevoir que la version électronique du dossier). Cette dernière doit être en tout point identique à la version électronique.

# TITRE

**Application desktop de gestion des heures de travail et facturation pour travailleur freelance.**

# MATÉRIEL ET LOGICIEL À DISPOSITION

1. Ordinateur type laptop « Corporate »
2. Visual Studio Codium
3. Git bash

# PRÉREQUIS

Connaissances en :

1. HTML
2. CSS
3. JavaScript
4. Node.js
5. React
6. ElectronJS
7. Figma
8. SQLite

# DESCRIPTIF DU PROJET

Application desktop de gestion de temps de travail et de facturation pour un travailleur freelance.  
L’application doit servir à noter les heures de travail de chaque projet pour tous les clients.  
L’utilisateur doit pouvoir :

* Ajouter des clients
* Ajouter des projets pour chaque client.
* Visualiser les clients et leur projets relatifs.
* Visualiser les projets et les heures de travail liées au projet.
* Consulter le calendrier en différentes vues (ex. jour, semaine, mois, année).
* Ajouter des heures de travail (timbrage) dans le calendrier, en liant un projet.
* Exporter les heures de travail (ex. format PDF - html) de chaque mois pour les clients en citant tous les projets et les relatives heures de travail.
* Sauvegarder toutes les données dans une base de données.

# LIVRABLES

Le candidat est responsable de livrer à son chef de projet et aux deux experts :

* Une planification initiale
* Un rapport de projet
* Un journal de travail

# POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

La grille d’évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, …).

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (Point A14 à A20) :

1. Création d’une maquette pour l’application (figma)
2. Gestion du code :
   1. Code organisé en dossiers et sous dossiers selon les fonctionnalités implémentées
   2. Code stocké dans un repository git avec commit fréquents, et messages de commit clairs et explicites
3. Respect des conventions de codage :
   1. Code non dupliqué et divisé en composants
   2. Code orienté objet,
   3. Code (composant) à responsabilité unique
   4. Code auto-documenté (nom de classe et méthode à signification business/fonctionnelle)
4. L’application adaptée pour des résolution d’écran multiples
5. L’application comporte des listes avec option de recherche
6. L’application comporte un système de notification afin de fournir un feedback à   
   l’utilisateur pour donner suite à une action de sa part
7. Le processus de compilation et déploiement est bien documenté et fonctionnel

# VALIDATION

|  |  |  |
| --- | --- | --- |
|  | Lu et approuvé le : | Signature : |
| Candidat : |  |  |
| Expert n°1 : |  |  |
| Expert n° 2 : |  |  |
| Chef de projet : |  |  |