|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | **Informaticien/-ne CFC**  Travail pratique individuel 2025 (TPI) |  | |
|  | |

Modèle de rapport v1.4

Candidat No 151464

Nom du candidat : Max Langenegger

Elevate – Projet TPI

### Sommaire

Elevate – Projet TPI 0

1 Résumé du rapport du TPI 2

1 Les grandes lignes du projet 3

1.1 Analyse de la situation initiale 3

1.2 Analyse de l’état désiré 3

1.3 Cahier des charges / exigences du système 3

1.4 Organisation du projet 3

2 Analyse préliminaire 4

1.1 Objectifs du système 4

2.1 Analyse de risque 4

1.2 Sécurité de l’information et protection des données 4

2 Concept 5

2.1 Exigences du système 5

2.2 Architecture du système 5

2.3 Plan d’intégration des systèmes 5

2.4 Concept d‘implémentation 5

2.5 Concept de formation 5

2.6 Concept de tests 5

2.7 Moyens nécessaires 5

3 Réalisation 6

3.1 Spécifications détaillées 6

3.2 Design du système 6

3.3 Configuration xyz 6

3 Test 7

3.1 Procédure de test 7

3.2 Protocol de test 7

3.3 Signature du protocole de test 7

3.4 PWA mode hors-ligne 7

3.5 Vérification connexion réseau PWA 9

3.6 Vérification connexion réseau PWA 10

4 Tests Techno 11

4.1 Test PWA 11

3.1 PWA mode hors-ligne 13

5 Conclusion 14

5.1 Améliorations possibles 14

5.2 Auto-évaluation 14

4 Bibliographie: liste des sources et références 15

5 Glossaire 16

6 Signatures 17

7 Annexes 18

# Résumé du rapport du TPI

Le Résumé du rapport du TPI est une présentation conceptuelle du travail effectué et du résultat attendu d’au maximum une page A4 qui permettent au lecteur une rapide compréhension du rapport de travail. . Il contient trois paragraphes : Situation de départ, mise en œuvre, résultats.

Pour plus de détails, veuillez vous référez au Manuel ICT - partie B : Documentation / rapport du TPI - Question 14 - page 81.

# Les grandes lignes du projet

## Analyse de la situation initiale

Quelle est la situation actuelle? Vous pouvez éventuellement illustrer le processus.

## Analyse de l’état désiré

A quoi doit ressembler la solution/le produit fini? Vous pouvez éventuellement illustrer le nouveau processus.

## Cahier des charges / exigences du système

Cahier des charges détaillé découlant du mandat de projet. Définition exacte et mesurable des objectifs à atteindre.

Décrire et structurer les exigences du système désiré.

## Organisation du projet

Pour l’organisation de mon projet, j’ai choisi d’utiliser un planning en mode cascade qui se trouvera dans un fichier Excel disponible dans les annexes. Une feuille *Journal de bord* sera aussi présente dans ce classeur Excel qui servira à lister les heures passés avec la description des tâches effectuées.

Plusieurs sauvegardes du projet sont faîtes quotidiennement, une se trouve en local dans les fichiers de mon téléphone, une version est disponible sur mon [github](https://github.com/langeneggerm/TPI_Elevate) et une autre version se trouve sur une clé USB personnel. Ces trois supports permettent d’avoir plusieurs issues de secours si mon environnement de travail qui est OneDrive lâche.

Citer la méthode de gestion de projet utilisée (Méthode en phases, agile, scrum, …)

Etablir la liste des participants au projet avec leur rôle respectif.

Décrire comment sont gérées les sauvegardes de la documentation et du code (fréquence, supports utilisés, utilisation de logiciel de gestion de version, …)

# Analyse préliminaire

## Objectifs du système

### Analyse de l’état actuel

Quelle est la situation actuelle? Vous pouvez éventuellement illustrer le processus.

### Analyse de l’état désiré

A quoi doit ressembler la solution/le produit fini? Vous pouvez éventuellement illustrer le nouveau processus.

### Processus d’entreprise concernés

Quels processus métier sont concernés par le mandat du projet?

### Objectifs

Liste détaillée des objectifs mesurables devant être atteints découlant du mandat de projet.

## Analyse de risque

Quels risques sont engendrés en cas de non réalisation du projet? Que se passe-t-il en cas d’échec du projet ? Quels sont les risques les plus importants du projet ?

## Choix de variante

Le premier choix que j’ai dû faire concerne le système de connexion que les commissaires vont utiliser pour se connecter à leurs comptes pour qu’ils puissent entrer des résultats dans la base de données.

Le dilemme était de choisir entre un mot de passe ou un code PIN. J’ai choisi le mode de connexion avec le code PIN, qui sera plus utile à entrer lors de la connexion pour chaque personnes sur le terrain grâce à sa rapidité d’exécution.

## Sécurité de l’information et protection des données

Quelles sont les menaces/dangers pour la protection des données et des systèmes et par quelles mesures peuvent-ils être contournés/supprimés. Les données sont-elles soumises à la protection des données?

# Concept

## Exigences du système

## Architecture du système

## Plan d’intégration des systèmes

## Concept d‘implémentation

## Concept de formation

## Concept de tests

Comment les tests sont-ils effectués? Blackbox, Whitebox … .

## Moyens nécessaires

Conseil: D’autres chapitres peuvent s’ajouter à cette partie en fonction du projet.

# Réalisation

## Spécifications détaillées

## Design du système

## Configuration xyz

Conseil: D’autres chapitres peuvent s’ajouter à cette partie en fonction du projet.

# Test

## Procédure de test

## Protocol de test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. | Objet testé | Description du test | Attente | Résultat | Visa |
| 1.0 |  |  |  |  |  |

## Signature du protocole de test

|  |  |  |
| --- | --- | --- |
| Date | Nom | Signature |
|  |  |  |

## PWA mode hors-ligne

### Procédure de test

*index.html*

### Protocol de test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. | Objet testé | Description du test | Attente | Résultat | Visa |
| 1.0 |  |  |  |  |  |

### Signature du protocole de test

|  |  |  |
| --- | --- | --- |
| Date | Nom | Signature |
|  |  |  |
|  |  |  |

## Vérification connexion réseau PWA

### Procédure de test

Tests technologiques pour l'enregistrement d'une image dans une base de données MySQL et la récupération d'une image depuis une base de données afin de l'afficher dans une page HTML.

### Protocol de test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. | Objet testé | Description du test | Attente | Résultat | Visa |
| 1.0 |  |  |  |  |  |

Protocole de test détaillé avec Visa.

### Signature du protocole de test

|  |  |  |
| --- | --- | --- |
| Date | Nom | Signature |
|  |  |  |

## Vérification connexion réseau PWA

### Procédure de test

### Protocol de test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. | Objet testé | Description du test | Attente | Résultat | Visa |
| 1.0 |  |  |  |  |  |

Protocole de test détaillé avec Visa.

### Signature du protocole de test

|  |  |  |
| --- | --- | --- |
| Date | Nom | Signature |
|  |  |  |

# Tests Techno

## Test PWA

Tests technologiques pour la mise en place d'une application PWA de base. Ce test permettra de créer un site avec un manifest permettant l’installation du site en tant qu’application PWA basique sans aucune autre fonctionnalité.  
  
J’ai commencé par créer un simple fichier html qui possède un lien vers un manifest ainsi qu’un texte *Hello World PWA*.

<html>

<head>

    <link *rel*="manifest" *href*="manifest.json" />

</head>

<body>

Hello World PWA

</body>

</html>

J’ai aussi créé le manifest permettant l’installation de l’application avec différents paramètres tels que le nom de l’application, l’url de l’application et les icones qui seront affiché sur l’application

{

    "name": "Test PWA",

    "short\_name": "PWA",

    "description": "Ce site est un site de test pour une PWA",

    "start\_url": "/testTechno/PWASimple",

    "display": "standalone",

    "background\_color": "#ffffff",

    "theme\_color": "#a70d0d",

    "icons": [

        {

            "src": "bob\_192x192.png",

            "sizes": "192x192",

            "type": "image/png"

        },

        {

            "src": "bob\_512x512.png",

            "sizes": "512x512",

            "type": "image/png"

        }

    ]

}

Lors de l‘accès au [site](https://elevate.emf-infopro-tpi.ch/testTechno/PWASimple/), celui-ci nous demande bien si l’on veut installer le site en tant qu’application.

Une image contenant texte, capture d’écran, Police, logo

Le contenu généré par l’IA peut être incorrect.Une image contenant texte, capture d’écran, Système d’exploitation, Page web

Le contenu généré par l’IA peut être incorrect.

L’application est fonctionnelle et téléchargeable sur navigateur ET aussi téléphone mobile.

Une image contenant texte, capture d’écran, dessin humoristique, graphisme

Le contenu généré par l’IA peut être incorrect.

## PWA mode hors-ligne

Tests technologiques pour faire fonctionner une application PWA en mode hors ligne (cache des fichiers). Il faut ajouter un serviceWorker qui va gérer les fichiers que le navigateur va mettre en cache. Cela permet de rendre l’application disponible même hors réseau.

Pour permettre l’ajout et l’utilisation d’un serviceWorker, j’ai d’abord créé un lien entre le fichier html et le fichier sw.js. J’ai ajouté un bout de JavaScript dans le html qui va vérifier que le navigateur supporte les serviceWorker et enregistre sw.js si le navigateur est correct.

    <script *type*="text/javascript" *src*="sw.js"></script>

    <script>

        if ('serviceWorker' in navigator) {

      window.addEventListener('load', () => {

        navigator.serviceWorker.register('/testTechno/PWAHorsLigne/sw.js')

          .then(*registration* => {

            console.log('Service Worker enregistré avec succès:', *registration*);

          })

          .catch(*error* => {

            console.log('Erreur lors de l’enregistrement du Service Worker:', *error*);

          });

      });

    }

    </script>

*sw.js*

const CACHE\_NAME = 'v1';

const FILES\_TO\_CACHE = [

  '/favicon.ico',

  '/testTechno/PWAHorsLigne/',

  '/testTechno/PWAHorsLigne/index.html',

  '/testTechno/PWAHorsLigne/sw.js',

  '/testTechno/PWAHorsLigne/logo/bob\_192x192.png',

  '/testTechno/PWAHorsLigne/logo/bob\_512x512.png',

  '/testTechno/PWAHorsLigne/parapente.jpg',

  '/testTechno/PWAHorsLigne/manifest.json'

];

self.addEventListener('install', *event* => {

*event*.waitUntil(

    caches.open(CACHE\_NAME)

      .then(*cache* => {

        console.log('Mise en cache des fichiers');

        return *cache*.addAll(FILES\_TO\_CACHE);

      })

  );

});

self.addEventListener('activate', *event* => {

*event*.waitUntil(

    caches.keys().then(*keys* =>

*Promise*.all(

*keys*.filter(*key* => *key* !== CACHE\_NAME)

            .map(*key* => caches.delete(*key*))

      )

    )

  );

});

self.addEventListener('fetch', *event* => {

*event*.respondWith(

    caches.match(*event*.request)

      .then(*response* => *response* || fetch(*event*.request))

  );

});

Maintenant l’application est disponible et visible même lorsque la connexion est interrompue.

Dans un premier temps sur navigateur, la page affiche que nous sommes hors connexion.

Une image contenant dessin humoristique, smiley, sourire, émoticône

Le contenu généré par l’IA peut être incorrect.

Puis la page et les fichiers présent dans le cache s’affiche correctement et tous les fichiers sont bien présents dans le cache du navigateur :

Une image contenant capture d’écran, texte, Logiciel multimédia, Logiciel de graphisme

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, nombre, Police, ligne

Le contenu généré par l’IA peut être incorrect.

## PWA statuts réseau

Tests technologiques pour établir si une connexion réseau est active ou non dans une application PWA. Analyser les moyens possibles pour savoir si une connexion est perdue ou rétablie.

Pour effectuer ce test j’ai ajouté un fichier *main.js* qui permet de vérifier l’état actuel du réseau pour ensuite afficher un message différent en fonction de celui-ci

$(document).ready(function () {

    document.getElementById('status').innerHTML = navigator.onLine ? 'online' : 'offline';

    var target = document.getElementById('target');

});

function handleStateChange() {

    var timeBadge = **new** *Date*().toTimeString().split(' ')[0];

    var newState = document.createElement('p');

    var state = navigator.onLine ? 'online' : 'offline';

    newState.innerHTML = '' + timeBadge + ' Connection state changed to ' + state + '.';

    console.log(newState.innerHTML);

    target.appendChild(newState);

}

window.addEventListener('online', handleStateChange);

window.addEventListener('offline', handleStateChange);

Une image contenant texte, capture d’écran, oiseau

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran

Le contenu généré par l’IA peut être incorrect.

## Envoi et réception d’image

Tests technologiques pour l'enregistrement d'une image dans une base de données MySQL et la récupération d'une image depuis une base de données afin de l'afficher dans une page HTML.

Pour effectuer ce test j’ai utilisé deux fichiers php, un qui va s’occuper d’envoyer les images et d’afficher du HTML et un autre qui va récupérer les images dans la base de données.

*Upload\_base64.php*

<?php

$host = 'localhost';

$dbname = 'db\_tt';

$username = 'root';

$password = '';

try {

    $pdo = new *PDO*("mysql:host=$host;dbname=$dbname", $username, $password);

} catch (*PDOException* $e) {

    die("Erreur de connexion : " . $e->getMessage());

}

if (isset($\_FILES['image'])) {

    $imageData = file\_get\_contents($\_FILES['image']['tmp\_name']);

    $base64Image = base64\_encode($imageData);

    $stmt = $pdo->prepare("INSERT INTO t\_img (img) VALUES (?)");

    $stmt->bindParam(1, $base64Image);

    $stmt->execute();

    echo "Image encodée en base64 enregistrée avec succès dans `t\_img` !";

}

?>

<!DOCTYPE *html*>

<html>

<head>

    <meta *charset*="UTF-8">

    <title>Envoyer une image (base64)</title>

</head>

<body>

    <h1>Uploader une image en base64</h1>

    <form *action*="upload\_base64.php" *method*="post" *enctype*="multipart/form-data">

        <input *type*="file" *name*="image" *required*>

        <input *type*="submit" *value*="Envoyer">

    </form>

    <p><a *href*="view\_base64.php">Voir les images enregistrées</a></p>

</body>

</html>

Le fichier upload va d’abord créer une connexion à la base de données et ensuite vérifier si il y a bien une image a envoyer pour ensuite l’encoder en base64 et l’ajouter à la base de données.  
Les deux fichiers php vont envoyer et recevoir les images sur la base de données stockée en local grâce à WAMP

*View\_base64.php*

<?php

$host = 'localhost';

$dbname = 'db\_tt';

$username = 'root';

$password = '';

try {

    $pdo = new *PDO*("mysql:host=$host;dbname=$dbname", $username, $password);

} catch (*PDOException* $e) {

    die("Erreur de connexion : " . $e->getMessage());

}

$stmt = $pdo->query("SELECT img FROM t\_img");

$images = $stmt->fetchAll(*PDO*::FETCH\_ASSOC);

?>

<!DOCTYPE *html*>

<html>

<head>

    <meta *charset*="UTF-8">

    <title>Images enregistrées (base64)</title>

</head>

<body>

    <h1>Images dans la base (base64)</h1>

    <?php foreach ($images as $img): ?>

        <div *style*="margin-bottom:20px;">

            <img *src*="data:image/jpeg;base64,<?= htmlspecialchars($img['img']) ?>" *style*="max-width:300px;">

        </div>

    <?php endforeach; ?>

</body>

</html>

View va s’occuper de récupérer les images et de les afficher sur la page actuelle.

Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, dessin humoristique, art

Le contenu généré par l’IA peut être incorrect.

## Requêtes en attentes et base de données locale

Tests technologiques sur les moyens d'enregistrer des données (les résultats en attente) en local et de gérer ces données : cache, base de données locale, fichier, autre.

*index.html*

<!DOCTYPE *html*>

<html *lang*="fr">

<head>

  <meta *charset*="UTF-8">

  <title>PWA Images</title>

  <link *rel*="manifest" *href*="manifest.json" />

  <script *type*="text/javascript" *src*="sw.js"></script>

  <script *src*="js/db.js"></script>

  <script *src*="js/main.js"></script>

  <script>

    if ('serviceWorker' in navigator) {

      window.addEventListener('load', () => {

        navigator.serviceWorker.register('/testTechno/PWARequeteDonneesCache/sw.js')

          .then(*registration* => {

            console.log('Service Worker enregistré avec succès:', *registration*);

          })

          .catch(*error* => {

            console.log('Erreur lors de l’enregistrement du Service Worker:', *error*);

          });

      });

    }

  </script>

</head>

<body>

  <h1>PWA - Gestion des images</h1>

  <div *id*="etatConnexion" *style*="padding: 10px; font-weight: bold;"></div>

  <input *type*="file" *id*="imageFile" *accept*="image/\*">

  <button *id*="uploadBtn">Envoyer l’image</button>

  <button *id*="loadBtn">Charger les images</button>

  <div *id*="resultat"></div>

</body>

</html>

Le fichier html contient les divers liens vers les fichiers JavaScript ainsi qu’un input, deux boutons et un div qui va accueillir le résultat des fichiers php.

*main.js*

  document.addEventListener('DOMContentLoaded', () => {

async function envoyerImage() {

  const fileInput = document.getElementById('imageFile');

  const file = fileInput.files[0];

  if (!file) return;

  const formData = **new** *FormData*();

  formData.append('image', file);

  if (navigator.onLine) {

    try {

      const res = await fetch('upload\_base64.php', {

        method: 'POST',

        body: formData

      });

      if (res.ok) {

        alert(await res.text());

      } else {

        console.error('Réponse non OK, stockage local.');

        await saveImageOffline(file);

      }

    } catch (err) {

      console.error('Erreur réseau, stockage local.');

      await saveImageOffline(file);

    }

  } else {

    console.log('Hors ligne, image stockée localement.');

    alert("Les images seront envoyé lorsque vous serez connecté à internet");

    await saveImageOffline(file);

  }

}

  async function chargerImages() {

    if (navigator.onLine) {

      console.log("chargement en ligne")

      fetch('view\_base64.php')

        .then(*res* => *res*.json())

        .then(*data* => {

          const container = document.getElementById('resultat');

          container.innerHTML = '';

*data*.forEach(*img* => {

            const image = document.createElement('img');

            image.src = `data:image/jpeg;base64,${*img*}`;

            image.style.width = '200px';

            image.style.margin = '10px';

            container.appendChild(image);

          });

        });

    } else {

      try {

        console.log("chargement hors ligne")

        const images = await getAllImagesOffline();

        const container = document.getElementById('resultat');

        container.innerHTML = '';

        images.forEach(*img* => {

          const image = document.createElement('img');

          const reader = **new** *FileReader*();

          reader.onloadend = function() {

            image.src = reader.result;

            image.style.width = '200px';

            image.style.margin = '10px';

            container.appendChild(image);

          };

          reader.readAsDataURL(*img*);

        });

      } catch (err) {

        console.log('Erreur lors du chargement des images locales:', err);

      }

    }

  }

  // Boutons JS

  document.getElementById('uploadBtn').addEventListener('click', envoyerImage);

  document.getElementById('loadBtn').addEventListener('click', chargerImages);

  function mettreAJourEtatConnexion() {

const etat = document.getElementById('etatConnexion');

if (navigator.onLine) {

  etat.textContent = '🟢 Connecté à Internet';

  etat.style.color = 'green';

} else {

  etat.textContent = '🔴 Hors ligne';

  etat.style.color = 'red';

}

}

mettreAJourEtatConnexion();

window.addEventListener('online', () => {

mettreAJourEtatConnexion();

console.log('Connexion retrouvée. Envoi des images locales.');

syncImages();

});

window.addEventListener('offline', () => {

mettreAJourEtatConnexion();

console.log('Perte de connexion');

});

});

Ce fichier va principalement vérifier l’état du réseau et sélectionner les différentes méthodes à utiliser en fonction de l’état de celui-ci. Ce que l’on veut avec ce test c’est stocker les requêtes que l’on fait lorsque l’on est sans connexion pour ensuite les envoyer quand on récupère

# Conclusion

Conclusion sur le projet en général. Les objectifs sont-ils atteints ? Si non, pourquoi ?

## Améliorations possibles

Y a-t-il des améliorations possibles (nouvelles fonctionnalités, amélioration et/ou modification du design, modification du support de données…) ?

## Auto-évaluation

Quels sont mes sentiments vis-à-vis du travail effectué. Enoncé des facteurs qui expliquent le succès ou l’échec de la réalisation des objectifs du projet.

# Bibliographie: liste des sources et références

Conseil: Cette partie peut contenir une liste des sources et éléments littéraires utilisés.

# Glossaire trier par ordre alphabetique

|  |  |
| --- | --- |
| Terme | Signification |
|  |  |
|  |  |
|  |  |

# Signatures

Je soussigné déclare que les informations contenues dans ce rapport de travail pratique individuel rendu ce jour le XX.XX.20XX dans le cadre de la procédure de qualification de mon CFC d’informaticien/-ne, ne sont pas plagiées. Toutes les informations de sources extérieures ainsi que les informations fournies par des tiers durant le déroulement du travail sont consignées.

L’apprenti/-e doit signer la documentation avant la remise afin de témoigner de la justesse des informations qui y figurent.

|  |  |  |
| --- | --- | --- |
| Date | Nom | Signature |
|  |  |  |

# Annexes

Code imprimé, protocole de séances, etc.: spécifier complètement et annexer à la documentation.

Listings de scripts et programmes. Le travail fourni par la candidate ou le candidat doit être entièrement documenté. Omettre le code généré automatiquement s’il n’est pas nécessaire à la compréhension.

Dans le cas où des manuels ont été créés, vous pouvez les ajouter ici en tant qu’annexe.