

Projet Informatique: FIPA 1A



FLEM

**For Last emergency
Message**

BONNETON Emilien, HERNOT François, DESCLOUX Maxime

RAPPORT PROJET INFORMATIQUE FLEM

Sommaire :

I. Introduction

- A. Contexte du projet
- B. Objectifs du projet
- C. Structure du rapport

II. Configuration du serveur maison

- A. Choix du matériel et des logiciels
- B. Configuration du réseau
- C. Installation des services nécessaires (Apache, PHP, MySQL, etc.)
- D. Sécurisation du serveur (pare-feu, certificats SSL, etc.)

III. Développement du site web

- A. Conception de l'architecture du site
- B. Choix des technologies (PHP, HTML, CSS, JavaScript)
- C. Création des pages et fonctionnalités principales
- D. Intégration du design (CSS, mise en page)
- E. Gestion des interactions côté client (JavaScript)

IV. Hébergement du site web

- A. Configuration du serveur web (Apache)
- B. Déploiement du site web sur le serveur
- C. Gestion de la base de données (MySQL)
- D. Tests et vérifications de bon fonctionnement

V. Résultats et discussions

- A. Présentation du site web développé
- B. Analyse des problèmes rencontrés et des solutions adoptées

VI. Conclusion

- A. Récapitulation des objectifs atteints
- B. Réflexions sur les enseignements tirés du projet
- C. Perspectives d'amélioration et de développement futur

VII. Références

- A. Liste des sources consultées
- B. Liens vers la documentation utilisée

VIII. Annexe

- A. Captures d'écran du site web
- B. Code source du site web

I. Introduction

1. Contexte du projet:

Pour le projet informatique, nous souhaitons proposer l'application FLEM (For Last Emergency Message ou Francis Lalanne est mort, à vous de choisir) . Le projet propose une solution qui permettra aux utilisateurs de rentrer en sécurité chez eux.

2. Objectifs du projet:

L'objectif de notre application est de mettre en place tous les éléments d'urgence pour rentrer en sécurité chez soi.

La fonctionnalité principale de notre application est la mise en place d'un système de minuteur à partir duquel si la personne n'est pas arrivée au point d'arrivée défini et/ou dévie trop du trajet, le téléphone lancera une alerte aux personnes concerné.

Grâce à cette application mobile, les utilisateurs pourront aussi :

- Obtenir un itinéraire sûr pour rentrer chez eux.
- Définir des contacts à prévenir automatiquement lorsqu'on arrive bien chez soi ou lorsque l'on n'est pas arrivé dans un délai raisonnable à destination.
- Disposer d'un bouton alerte lorsqu'ils sont en danger, les contacts prédéfinis ainsi que les autres utilisateurs à proximité pourront recevoir la position GPS en direct de la personne.
- Signaler leur arrivée sur un lieu où ils sont en sécurité grâce à des lieux prédéfinis et aux points GPS de la personne.

3. Structure du rapport:

Nous commencerons ce rapport avec la partie technique qui contient la configuration du serveur maison, le développement du site web ainsi que son hébergement. Nous verrons ensuite la partie résultat du site web qui comporte la présentation du site, l'analyse de la performance du serveur ainsi que la comparaison avec d'autres solutions d'hébergement. Nous finirons cette partie par les problèmes que nous avons rencontrés lors du projet et les solutions que nous avons apportées à ces problèmes. Nous terminerons ce rapport par une conclusion qui contient tout d'abord un récapitulatif des objectifs atteints, une réflexion sur les enseignements tirés du projet ou encore les perspectives d'amélioration et de développement futur de notre site web.

II. Configuration du serveur maison

La complexité et la lenteur du service de mise en place de VMs nous ont forcé à mettre en place un serveur maison.

1. Choix du matériel et des logiciels

Le choix de WAMP (Windows, Apache, MySQL, PHP) comme environnement de développement pour configurer votre serveur maison et héberger votre site web comporte plusieurs justifications importantes :

Facilité d'installation : WAMP propose un package tout-en-un qui facilite grandement l'installation et la configuration des principaux composants nécessaires pour exécuter un serveur web, notamment Apache, PHP et MySQL. Il évite ainsi la nécessité d'installer et de configurer manuellement chaque composant séparément, ce qui peut être complexe pour les utilisateurs moins expérimentés.

Compatibilité avec Windows : Si vous utilisez le système d'exploitation Windows sur votre serveur maison, WAMP est un choix naturel, car il est spécifiquement conçu pour fonctionner sur cette plateforme. Il assure une compatibilité optimale avec les fonctionnalités et les paramètres de Windows, facilitant ainsi l'intégration avec le système d'exploitation.

Stabilité et performance : WAMP est largement utilisé et bénéficie d'une communauté active de développeurs. Les composants qu'il intègre, tels qu'Apache, PHP et MySQL, ont été éprouvés et optimisés pour offrir une performance stable et fiable. De plus, WAMP fournit des outils de gestion et de surveillance pour assurer un bon fonctionnement du serveur.

Prise en charge de MySQL : MySQL est l'un des systèmes de gestion de bases de données les plus populaires et puissants. En utilisant WAMP, vous bénéficiez d'une intégration transparente de MySQL, ce qui facilite la création et la gestion de votre base de données pour votre site web.

Flexibilité pour le développement web : WAMP offre une plateforme flexible pour le développement web, prenant en charge les technologies clés telles que PHP, HTML, CSS et JavaScript. Il fournit un environnement de développement local où vous pouvez facilement tester et déboguer votre site web avant de le déployer sur un serveur en ligne.

Communauté et ressources disponibles : WAMP bénéficie d'une communauté active d'utilisateurs et de développeurs qui fournissent un support technique, des tutoriels et des ressources en ligne. Cela facilite l'apprentissage et la résolution des problèmes éventuels rencontrés lors de la configuration du serveur et du développement du site web.

En combinant ces avantages, WAMP se présente comme une solution pratique, complète et conviviale pour configurer un serveur maison et héberger un site web. Cependant, il est

RAPPORT PROJET INFORMATIQUE FLEM

important de noter qu'il existe d'autres alternatives telles que LAMP (Linux, Apache, MySQL, PHP) ou MAMP (Mac, Apache, MySQL, PHP), qui peuvent être plus adaptées selon vos préférences et votre environnement de travail.

2. Configuration du réseau

Pour permettre l'accès à distance au serveur depuis Internet via la Box Freebox, vous devez effectuer une redirection de port vers l'adresse IP locale du serveur. Voici les étapes pour configurer une Box Freebox :

1. Accédez à l'interface de configuration de la Box Freebox. Cela est fait en ouvrant un navigateur web et en entrant l'adresse IP de la Box (généralement 192.168.1.1) dans la barre d'adresse. Connectez-vous à l'aide de vos identifiants de connexion.
2. Recherchez la section "Redirection de ports" ou "NAT/PAT" dans les paramètres de la Box. Les termes exacts peuvent varier en fonction de la version du firmware de la Box.
3. Créez une nouvelle règle de redirection de port en spécifiant les détails suivants :
 - Protocole : Pour notre besoin on choisit TCP
 - Port externe : Dans notre cas 80 pour le site en HTTP et 443 pour le site en HTTPS
 - Adresse IP locale : On renseigne ici l'adresse IP locale, préalablement rendu statique, de notre serveur sur le réseau local.
 - Port interne : On indique ici le port interne sur lequel notre serveur écoute, dans notre cas ce sont les ports par défaut apache : 80 et 443.
4. Enregistrez la règle de redirection de port et appliquez les modifications.

Après avoir effectué ces étapes, les requêtes externes vers l'adresse IP publique de notre Box Freebox, sur les ports spécifiés (soit 80 et 443), seront redirigées vers l'adresse IP locale de notre serveur. Cela permettra aux utilisateurs d'accéder à notre site web hébergé sur le serveur depuis Internet.

III. Développement du site web

1. Conception de l'architecture du site

L'étape de conception de l'architecture du site a constitué le point de départ du développement. Nous avons défini la structure globale du site, en identifiant les différentes sections, pages et fonctionnalités qui étaient nécessaires pour respecter le cahier des charges fixé. Cette étape a permis de créer une vision claire de la façon dont les différentes parties du site se sont articulées et ont interagi entre elles. C'est également à cette étape que la répartition des tâches a été décidée.

2. Choix des technologies (PHP, HTML, CSS, JavaScript)

Pour le développement du site web, nous avons opté pour l'utilisation de technologies courantes telles que PHP, HTML, CSS et JavaScript. PHP a été utilisé pour la logique côté serveur, permettant de traiter les données et d'interagir avec la base de données. HTML a été utilisé pour la structure et le contenu des pages, tandis que CSS a été utilisé pour la mise en forme et le design. JavaScript est utilisé pour gérer les interactions côté client, offrant une expérience utilisateur dynamique et réactive.

3. Création des pages et fonctionnalités principales

Une fois l'architecture définie et les technologies choisies, nous avons procédé à la création des pages et fonctionnalités principales du site. Cela a inclus la mise en place des différentes sections du site, dans notre cas, l'accueil, l'espace utilisateur, la page itinéraire, etc.. Nous avons veillé à ce que chaque page soit conçue de manière à offrir une expérience utilisateur intuitive.

4. Intégration du design (Bootstrap :html, css, javascript)

Nous avons choisi d'utiliser Bootstrap pour développer notre site internet en raison de ses nombreux avantages en termes de conception et de design. Bootstrap est un framework front-end réputé qui offre une large gamme d'outils et de fonctionnalités pour créer une interface utilisateur moderne et attrayante. Grâce à sa grille de mise en page responsive, nous avons pu assurer une expérience utilisateur cohérente et optimisée sur différents appareils et résolutions d'écran. De plus, Bootstrap propose une bibliothèque complète de composants prédéfinis tels que des boutons, des formulaires, des menus déroulants, ce qui a permis de gagner du temps dans le développement et de garantir une apparence uniforme à travers toutes les pages du site. En utilisant Bootstrap, nous avons également bénéficié de son système de classes CSS préétablies, ce qui nous a permis de personnaliser facilement l'apparence et le style du site sans avoir à écrire de nombreuses lignes de code supplémentaires. En fin de compte, l'utilisation de Bootstrap nous a permis de créer un site

RAPPORT PROJET INFORMATIQUE FLEM

internet moderne, réactif et esthétiquement agréable, tout en optimisant notre processus de développement.

5. Gestion des interactions côté client (JavaScript)

JavaScript a été utilisé pour l'intégration et l'utilisation d'API afin d'enrichir les fonctionnalités du site. Ces API ont permis d'effectuer des opérations telles que l'intégration de services de géolocalisation, de création d'itinéraire, d'information concernant les horaires de transports, ou encore d'interactions en temps réel avec les utilisateurs. Cela a permis d'améliorer l'engagement des utilisateurs et de rendre le site plus convivial et interactif, en offrant des fonctionnalités avancées et en facilitant les échanges entre le site et les utilisateurs.

IV. Hébergement du site web

- A. Configuration du serveur web (Apache)**
- B. Déploiement du site web sur le serveur**
- C. Gestion de la base de données (MySQL)**
- D. Tests et vérifications de bon fonctionnement**

V. Résultats et discussions

1. Présentation du site web développé

Nous avons réussi à créer, après plusieurs essais et changements, un site web fonctionnel, esthétiquement attrayant et convivial, répondant aux besoins spécifiques de notre projet. Le site comprend la plupart des fonctionnalités principales prévues, telles que des espaces utilisateur et des fonctionnalités avancées comme le calcul d'itinéraire, l'envoi d'alerte, etc. présentation de chaque page:

- **Index:** cette page fait office de page d'accueil. Elle possède un court texte introductif illustré. Cela permet à nos visiteurs d'avoir une idée claire de notre solution, contexte enjeu objectif et rôle...

Toutes les pages contiennent une barre de navigation.

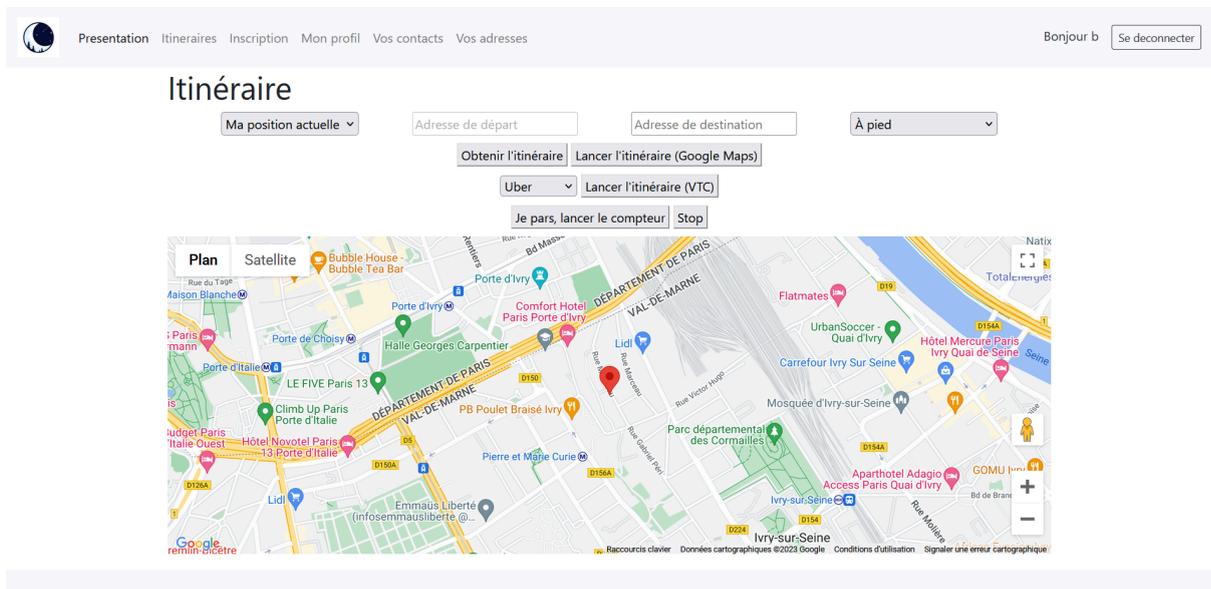
Elle permet de jongler entre les différents onglets depuis n'importe quelle page du site. Elle contient aussi une autre fonctionnalité importante: un onglet de connexion. Celui-ci est configuré en PHP pour que lorsque l'utilisateur se connecte ce bouton laisse place à un bouton de déconnexion. D'ailleurs pour faciliter la répliquabilité et la lisibilité du code. nous avons défini cette nav-bar dans une autre page (fonction.php) et nous lui faisons appel dans toutes les autres pages via la fonction include("fonction.php) puis les appel de la fonction "entete()" sur chaque page. De la même manière, nous avons défini une fonction pour le pied de page.



Affichage dynamique de l'utilisateur connecté

RAPPORT PROJET INFORMATIQUE FLEM

- **Itinéraire** : C'est la page qui contient la fonctionnalité centrale du projet. C'est sur cette page que l'on va pouvoir calculer son trajet, démarrer un itinéraire ou encore démarrer le compte à rebours qui permettra de s'assurer que l'on est bien rentré chez soi.



2. Analyse des problèmes rencontrés et des solutions adoptées

Nous avons rencontré de nombreux problèmes lors du développement de notre site web. Les principaux étaient liés à l'utilisation d'API. Pour exemple, nous avons tout d'abord choisi d'utiliser OpenStreetMap pour la partie map et itinéraire de notre site car OpenSource. Cependant lorsque nous avons voulu utiliser la bibliothèque Leaflet Routing Machine, qui devait permettre d'effectuer le calcul d'itinéraires, nous avons appris que celle-ci était défaillante. Nous nous sommes donc rabattu sur Google Map. Ensuite, nous voulions utiliser l'API RATP de Pierre Grimaud dans le but d'obtenir les horaires et itinéraires en train de la RATP et de la SNCF mais encore une fois, celle-ci était défaillante, nous recevions des messages d'erreur de communication avec le serveur lorsque nous faisons les essais. Nous nous sommes donc mis d'accord d'utiliser encore une fois l'API Google Map pour remplacer celle-ci.

Nous avons eu d'autres problèmes, non techniques cette fois, comme le départ d'un des membres de notre groupe. Ensuite, lors de la demande de création d'un VM à l'école, nous n'avons pas eu de réponse avant un certain temps et lorsque nous en avons, ce n'était absolument pas ce que nous attendions. Nous avons donc choisi d'être directement hébergés par Emilien.

VI. Conclusion

1. Récapitulation des objectifs atteints

A ce jour, le site dispose de pratiquement toutes les options que nous voulions intégrer mais ils en manquent encore certaines. La possibilité de prévenir les gens qui se situent à côté de nous lorsqu'une alerte se lance ou encore celle d'émettre des sons supposés effrayants sont absents à ce jour.

2. Réflexions sur les enseignements tirés du projet

Il est compliqué de réaliser un projet de cette ampleur en groupe. Effectivement, tout le monde ne part pas avec la même base et ne travaille pas de la même façon. Il faut cependant faire avec pour mener à bien le projet. De plus, il est compliqué de respecter le calendrier des tâches. Très souvent, les fonctionnalités désirées ne fonctionnent pas comme on le souhaiterait, on doit donc passer plus de temps dessus et on accumule du retard. De plus, on a tendance à ne pas prévoir assez de marge pour les imprévus comme les dysfonctionnement d'API dans notre cas qui retardent vraiment la réalisation du projet.

3. Perspectives d'amélioration et de développement futur

Aujourd'hui, le site est bien avancé et en ligne. Après une discussion avec tous les membres du groupe, il a été décidé que l'on essaierait de terminer les projets malgré la fin du module avec l'école. Il ne manque pas beaucoup de choses pour que l'on y arrive.

VII. Annexe

1. Captures d'écran du site web



The screenshot shows the top navigation bar of the website. On the left is a logo of a globe. The navigation menu includes: [Presentation](#), [Itinéraires](#), [Inscription](#), [Mon profil](#), [Vos contacts](#), [Vos adresses](#), and [ALERTE SOS](#). On the right, there is a user status area with the text "Veuillez vous connecter" and a "Se connecter" button.

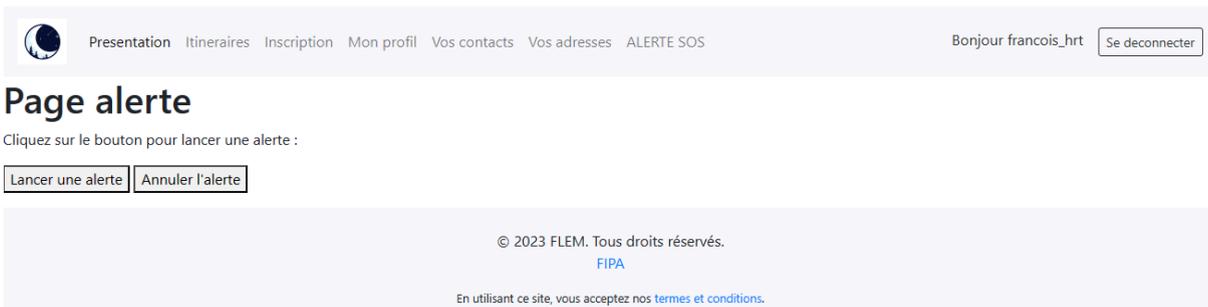
The main content area contains a paragraph of text on the left and a photograph on the right. The text discusses the importance of security in a modern society and describes the application's features, such as real-time monitoring and precise location tracking using GPS. The photograph shows a person in a dark, hooded jacket walking away from the camera down a city street.

Capture d'écran de la page d'accueil



The screenshot shows the login page. The navigation bar is identical to the home page. The main heading is "Connexion". Below it are two input fields: "Nom d'utilisateur:" and "Mot de passe:". To the right of the password field is a "Se connecter" button. Below the input fields are two buttons: "Inscription" and "Retour à la page d'accueil". At the bottom of the page, there is a copyright notice: "© 2023 FLEM. Tous droits réservés." with a link to "FIPA". Below that is a statement: "En utilisant ce site, vous acceptez nos [termes et conditions](#)."

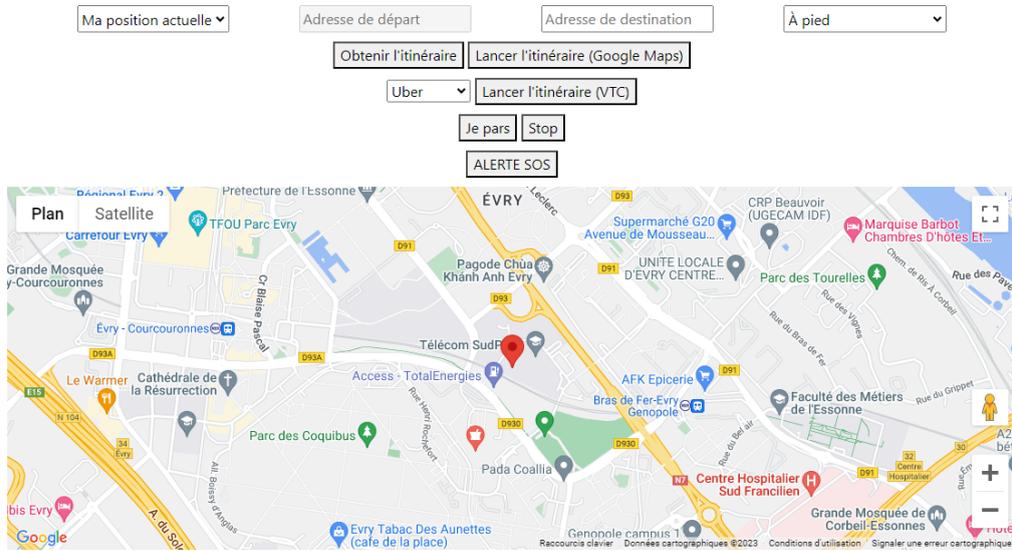
Capture d'écran de la page de connexion



The screenshot shows the alert page. The navigation bar is identical to the previous pages. On the right, the user is logged in, showing "Bonjour francois_hrt" and a "Se deconnecter" button. The main heading is "Page alerte". Below it is the instruction: "Cliquez sur le bouton pour lancer une alerte :". There are two buttons: "Lancer une alerte" and "Annuler l'alerte". At the bottom of the page, there is a copyright notice: "© 2023 FLEM. Tous droits réservés." with a link to "FIPA". Below that is a statement: "En utilisant ce site, vous acceptez nos [termes et conditions](#)."

RAPPORT PROJET INFORMATIQUE FLEM

Capture d'écran de la page disposant du bouton d'alerte



Capture d'écran de la page itineraire

Timer avant alerte : 1h 38min 46s

Le calcul d'itinéraires piétons est en bêta. Faites attention – Cet itinéraire n'est peut-être pas complètement aménagé pour les piétons.

25 Rue Charles Fourier, 91000 Évry-Courcouronnes, France

31,9 km. Environ 1 heure 19 min

Marcher jusqu'à Évry - Courcouronnes
Environ 13 minutes

Évry - Courcouronnes

Train vers Orry-La-Ville - Coye
10:17 - 10:29 (12 minutes, 4 arrêts)
Service assuré par RER

Juvisy

Train vers Saint-Quentin en Yvelines
10:43 - 10:55 (12 minutes, Un arrêt)
Service assuré par RER

Bibliothèque François Mitterrand

Marcher jusqu'à Bibliothèque François Mitterrand
Environ 3 minutes

Bibliothèque François Mitterrand

Bus vers Porte de Saint-Cloud
11:03 - 11:21 (18 minutes, 11 arrêts)
Service assuré par RATP

René Coty

Marcher jusqu'à 9 ter Rue d'Alésia, 75014 Paris, France
Environ 2 minutes

Capture d'écran du timer en cours avec l'itineraire sur la page itinéraire

2. Code source du site web

```
function initMap() {
  map = new google.maps.Map(document.getElementById('map'), {
    zoom: 15,
    center: { lat: 48.624507, lng: 2.442140 } // Coordonnées par défaut (Sydney, Australie)
  });

  directionsRenderer = new google.maps.DirectionsRenderer({
    panel: document.getElementById('directions-panel') // Affichage détaillé de l'itinéraire
  });
  directionsRenderer.setMap(map);

  // Autocomplétion d'adresses pour les champs de saisie
  var originSelect = document.getElementById('origin-select');
  var originInput = document.getElementById('origin-input');
  var destinationInput = document.getElementById('destination-input');
  var options = {
    types: ['geocode'] // Limitation des résultats à des adresses géocodables
  };
  var originAutocomplete = new google.maps.places.Autocomplete(originInput, options);
  var destinationAutocomplete = new google.maps.places.Autocomplete(destinationInput, options);

  // Initialiser le géocodeur
  geocoder = new google.maps.Geocoder();

  // Initialiser le service Directions
  directionsService = new google.maps.DirectionsService();
}
```

Capture d'écran code javascript

Dans cette partie du code JS, on fait appelle à différents API Google Maps pour initialiser la Maps de la page itinéraire.

```
function sendEmail() {
  // Récupérer les contacts pré-définis
  var contacts = ['contact1@example.com', 'contact2@example.com'];

  // Construire le corps de l'e-mail
  var subject = 'Alerte - Retard de trajet';
  var message = "Le temps de trajet estimé a été dépassé et l'utilisateur n'est pas arrivé à destination à l'heure prévue.";

  // Envoyer l'e-mail à chaque contact
  for (var i = 0; i < contacts.length; i++) {
    var contact = contacts[i];
    var mailtoLink = 'mailto:' + encodeURIComponent(contact) + '?subject=' + encodeURIComponent(subject) + '&body=' + encodeURIComponent(message);

    // Ouvrir le lien de messagerie dans une nouvelle fenêtre
    window.open(mailtoLink);
  }
}
```

Capture d'écran code Javascript 2

Dans cette partie du code, on peut voir la fonction sendEmail qui est à la charge de l'envoi d'un Email à des contacts rentrés dans le code directement. Cette fonction est appelée si le timer arrive à 0 sans que l'utilisateur soit arrivé.

Cette fonction est appelée à être modifiée. Par la suite, cette fonction fera appel à du PHP qui ira chercher les contacts pré-définis pas l'utilisateur et qui enverra un mail en toute autonomie sans intervention de l'utilisateur.

RAPPORT PROJET INFORMATIQUE FLEM

VIII. Références

A. Liste des sources consultées

B. Liens vers la documentation utilisée