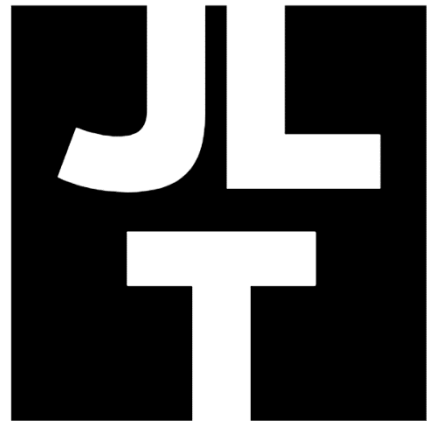


Bienvenue dans l'utilitaire vie scolaire

Choisissez ci-dessous l'action désirée

Générateur de mails →

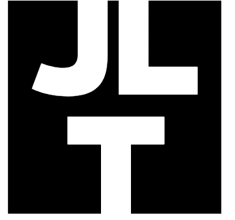
Gestion BDD



2022

Rapport De Projet

Outil mail vie scolaire – Jamai'l le temps





Tout a commencé en partant d'un constat !

*Chaque semaine,
toujours les mêmes mails
pour signaler des
absences*

Membres :

Saint Adrien La Salle – Lycée - Annapes

Eytane : eytane.debruyne@saintadrien.com

Alexandre : alexandre.colas@saintadrien.com

Noah : noah.dhaussy@saintadrien.com

Chaque année ce sont 77 mails d'absence qui sont individuellement rédigés et envoyés. Cela représente environ 55 000 caractères. Un français moyen rédige environ à une vitesse de 44 mots par minute et cela nous amène à un temps total de 690 minutes de rédaction sur une année soit 11h30 complètes de rédaction

Table de Contenu

1 • Introduction du projet	
A propos de nous	06
Objectif principal	08
Chronologie du projet	
2 • Etapes de brainstorming¹	10
Les multiples idées	
Expression du besoin par le personnel	
3 • L'idée finale	
Fonctionnalités	
Interface	
Logique	
4 • Mise en production	15
Message des développeurs	
L'interface (Ré – 1)	
Fonctionnalités et logique (Ré – 2)	
Nos outils à disposition	
Difficultés rencontrées	17
5 • Déploiement	19
Hébergement	
Implémentation et retours	
6 • Future	22
Client Feedback	23

Introduction

Du Projet

La solution au problème que nous résolvons devait être la moins tentaculaire possible pour mener à bien la production et respecter les dates limites qui nous ont été fixées !

Mise en œuvre :

Afin de conduire notre projet jusqu'au bout, nous avons opté pour des outils efficaces et simples. Certains déjà appris dans le cadre de la classe de NSI, d'autres nouveaux qui ont nécessité un investissement supplémentaire pour les prendre en main.

L'idée :

L'outil que nous proposons à l'issue de cette période de projet est un outil permettant de simplifier les envois de mails par la vie scolaire.

Les mails destinés à être remplacés par l'outil sont les mails d'absence simple (les plus fréquents)

Obstacles :

De nombreux obstacles ont été rencontrés durant la production et même durant les étapes de brainstorming. Ces obstacles seront détaillés dans une section spécifique.

Nous avons pu compter sur le fondement de la classe de NSI pour surmonter les problèmes : l'entraide.

Introduction

A propos de nous

Une équipe **créative** aux
talents hétéroclites.
Notre **force** !

Nous avons formé notre équipe avec des éléments de tous niveaux, avec comme unique idée en tête de rassembler des gens motivés et moteurs au projet.

Designer 3D, joueur de territorial.io, ou bien découpeur de palettes ; voilà notre multitude ! Chez JLT, la différence fait la force.

Notre vision

« Nos différences devraient nous aider au lieu de nous opposer. Pour ma part, là comme partout, je ne crois qu'aux différences, non à l'uniformité. Et, d'abord, parce que les premières sont les racines sans lesquelles l'arbre de liberté, la sève de la création et de la civilisation se dessèchent. »

Albert Camus





Qui sommes nous vraiment ?

Nous sommes tous trois élèves du lycée Saint Adrien, en classe de première, suivant la spécialité de NSI.

Nous avons choisi cette spécialité car elle nous paraissait unanimement attractive de par sa nouveauté et son programme. Nous ne regrettons pas ce choix car il nous amène ici, à la rédaction de ce rapport, rendant compte de notre projet fièrement conclu.



Eytane Debruyne
Web Designer / JS Developer



Alexandre Colas
Stagiaire qui a des idées



Noah Dhaussy
Joueur de territorial.io

Eytane Debruyne, 16 ans, développeur de sites, explore et connaît toutes les faces du web. Son implication plus que prédominante dans le projet nous a permis de finir le site ainsi que le rapport dans les temps impartis.

De plus ce projet lui a permis de mettre en usage ses facilités en informatique mais malheureusement pas d'utiliser ses compétences de designer 3D.

Alexandre Colas est un jeune passionné de meubles en palettes et d'impression 3D. Ces centres d'intérêts l'amènent à éprouver un averse intérêt pour les nouvelles technologies.

Doté d'une grande curiosité il n'aura pas hésité à chercher à comprendre des choses comme le code du projet par lui-même.

Noah Dhaussy, 16 ans, passionné d'informatique depuis son plus jeune âge, possède un esprit très vif et fait preuve d'imagination dans tous les domaines.

Son sérieux lui permet de ne jamais se décrocher du travail auquel il porte un très grand intérêt.



Objectif Principal et contexte

Nos principes :



Main dans la main

Avec les principaux intéressés. Nous sommes entrés en contact avec le public visé de l'outil pour écouter leur besoin



Solution soignée

Fournir un produit fonctionnel et agréable à utiliser à tout instant, peu importe l'avancement du projet



Excellent Teamwork₂

Un travail synchronisé et réparti grâce à une organisation à travers différents outils numériques.

L'objectif :

Utilité

Le but est que l'outil produit présente une réelle utilité pour le travail du CPE, cette utilité se traduira par la grande variété de mails pris en charge par l'outil.

Flexibilité

Pour garantir la longévité de l'outil il lui faut être flexible. Il faut que les modèles et valeurs prédéfinis puissent être modifiés facilement.

Facilité

Un outil hostile à l'égard de l'utilisateur ne sera pas un outil réflexe. Il est donc nécessaire que cette solution soit la plus abordable possible. C'est un aspect du développement qui s'appelle l'ux (l'expérience utilisateur)

Le contexte :

Le projet, en résumé

Ce projet s'étend sur l'entièreté du 3ème trimestre. Il débuta par la formation des groupes. Entre 2 et 4 par groupe.

La ligne directrice de ce projet était de produire un outil qui pourrait être d'une utilité quelconque à Saint-Adrien. Certains groupes sont partis sur la création de jeu vidéo, d'autres sur la conception d'une solution mêlant logiciel et matériel, et enfin, d'autres, comme nous, ont décidé de créer une interface remplissant une fonction spécifique.

Cette
entreprise est
née de nos
esprits
créatifs.



Notre Voyage

C'est après des semaines de réflexion, de recherche, de voyages périlleux dans l'enceinte de l'établissement, ainsi que de réunions improvisées chez les CPE que nous est venue cette idée.

Vous permettant de transformer **ces heures** de rédaction **en simples clics** pour vous faciliter la vie.

Déroulement du projet

- Mars** → **Recherche idées...**
Ce mois-ci le projet a été annoncé. Plusieurs idées nous sont venues avant de se consacrer à celles-ci
- Avril** → **Début de l'élaboration et présentation**
Nous produisons des schémas et maquettes pour la présentation du projet au groupe NSI.
- Mai** → **Début de la production**
Recherche des technologies à utiliser, début de création de la page web.
- **Création de la logique**
Programmation de la logique de l'outil en Javascript, basée sur nos flowcharts.
- **Mise en ligne**
Publication sur Github pour que les autres membres puissent consulter et collaborer.
- **Outil fonctionnel et démonstration**
Pour la démonstration technique aux professeurs il fallait qu'à ce stade l'outil soit utilisable
- Juin** → **Implémentation**
Intégration d'un raccourci sur école directe + brief des CPE sur l'outil pour l'intégrer dès la rentrée
- **Retour du rapport**
Retour du rapport fini, ainsi que de l'outil et son code

2

Etapes du

Brainstorming

Notre arrivée jusqu'à l'idée finale ne fut pas une ligne droite mais plutôt un chemin sinueux. Voici nos pistes abandonnées.

01

Appli satisfaction cantine

Point clé

- Application qui aurait envoyé les statistiques et les éventuels commentaires pour chaque repas au responsable cantine

Raison abandon

- Création d'une app hors de portée
- Solution répondant à un problème non existant

02

Solution de partage de fichiers

Point clé

- Page web pour échanger des fichiers entre profs et élèves de manière rapide et facile, le tout directement sur le réseau Saint Adrien

Raison abandon

- Projet nécessitant une connaissance de la partie serveur d'une page web, à un niveau élevé
- Une solution similaire existe déjà sur internet : snapdrop.net



L'expression du besoin Par l'équipe de CPE

« Ah oui ça pourrait être
pratique ! » Mme Diop

« Le gain de temps ne serai
pas de refus » M. Tiedrez

Lorsque l'idée de l'outil de génération mails d'absence nous est venue nous en avons parlé aux personnels de vie scolaire, qui ont réagi positivement à la solution née. L'idée finale était établie



3 à 4

CPE visés par
notre solution



+100h

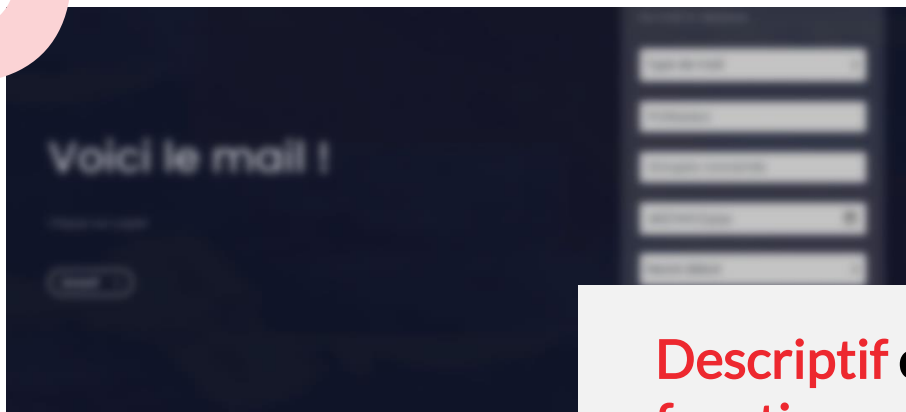
Economisées grâce à
notre solution



+1100

Mails seront désormais
gérés par notre outil

3 L'idée Finale



Il aura fallu **plus d'un mois entier** pour définir un projet final, mais une fois lancé **tout s'accéléra**

Interface

Idée finale de l'interface

L'idée finale pour l'interface a été celle d'une page web d'une simplicité maximum, nous avons hésité pour une interface en python mais ce n'était pas une solution adaptée pour le problème résolu par l'outil.

Logique

Idée finale pour la logique

Afin de créer l'algorithme donnant vie à notre projet, nous avons réalisé des flowchart₂ et des arborescences de réponses. Notre idée finale a été d'utiliser javascript pour son intime intégration aux pages web.

Descriptif et fonctionnalités

Bref descriptif

Notre outil permet de générer automatiquement les mails les plus redondants du CPE (cad. les mails de notification d'absence). Ceci en indiquant les informations clés du mail, le site se charge du reste pour économiser le temps. Il permet de rédiger les différents mails sur une base commune : un squelette de mail rédigé suivant une syntaxe et un style similaire à l'entièreté des mails des différents CPE

Les fonctionnalités clé

Nos exigences quant à notre outil sont les suivantes :

- Proposant plusieurs modèles (4.3)
 - Adaptés aux différentes situations
- Flexible et facile à maintenir (5.1)
 - Ajout ou modification simple des champs prédéfinis tels que les noms de CPE
- Rapide à accéder (4.2 - 5.1)
 - Hébergement stable et rapide + page optimisée
- Disponible à tout instant (5.1)
- Evolutif
 - Préparé à l'ajout de greffons et de nouvelles fonctionnalités

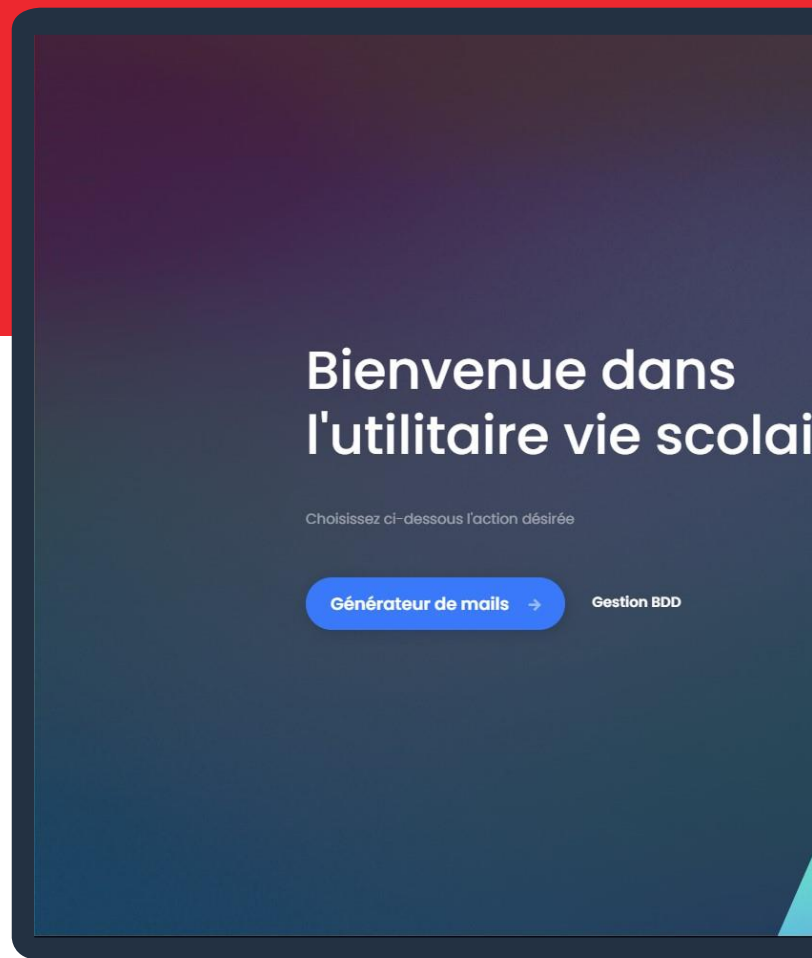
Mise en Production

Date de
mise en
production

26 avr. 2022

Point de départ Maquette

Message du développeur

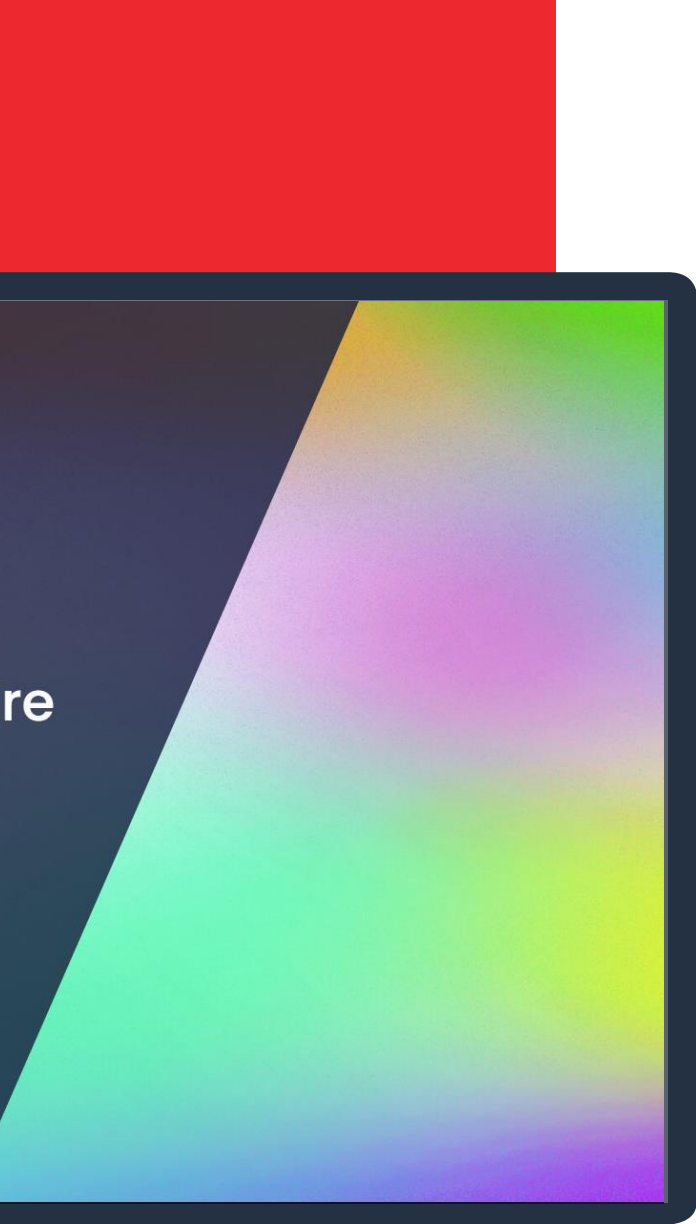


Interface finale de la page d'accueil

La création de cet outil ne fut pas simplement un réinvestissement de compétences, mais un réel approfondissement des différents sujets déjà étudiés dans le cadre de la classe de NSI. Les capacités remises en œuvre à l'occasion de ce projet sont celles des modules traitant des interactions homme / machine (Ré – 1), et de l'ensemble des modules abordant l'algorithmie (Ré – 2)

De nombreuses difficultés ont dû être surmontées pour conduire ce projet à son stade actuel.

Découvrons dans cette section du rapport le cheminement de la production numérique, en commençant par la programmation de l'interface.



Ré – 1

Debut de conception de l'interface

Pour créer notre interface nous avons choisi une page web HTML - CSS - JS reposant sur plusieurs frameworks³ et dépendances⁴ qui seront détaillés dans la suite de cette section sur l'interface

- 01 Création d'une maquette fonctionnelle pour la présentation au groupe NSI
- 02 Création des pages en HTML sans style
- 03 Liens entre pages et formulaire
- 04 Intégration des framework et du CSS

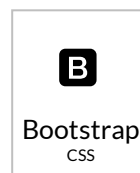
Proportions de technologies :



HTML5

9,5 %

mail-generator.html
landing-page.html



Bootstrap
CSS

25,7 %

../Bootstrap
../font-awesome
../css

1

Création de la maquette

Afin de définir la direction artistique de la page il nous fallait faire une maquette. Elle a été réalisée sur Figma, un site internet de prototypage permettant de réaliser des pages comme on le ferai sur Word en glissant déposant différents éléments sur la page.

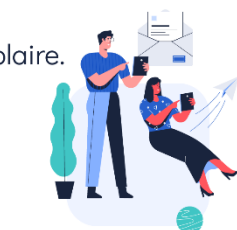
Seul bémol d'un tel outil : il est impossible de récupérer du code, il faut obligatoirement tout refaire à la main en html une fois la maquette finie.

Bienvenue dans
l'utilitaire vie scolaire.

Selectionnez ci dessous l'action désirée !

Modèles de Mail

Accès BDD



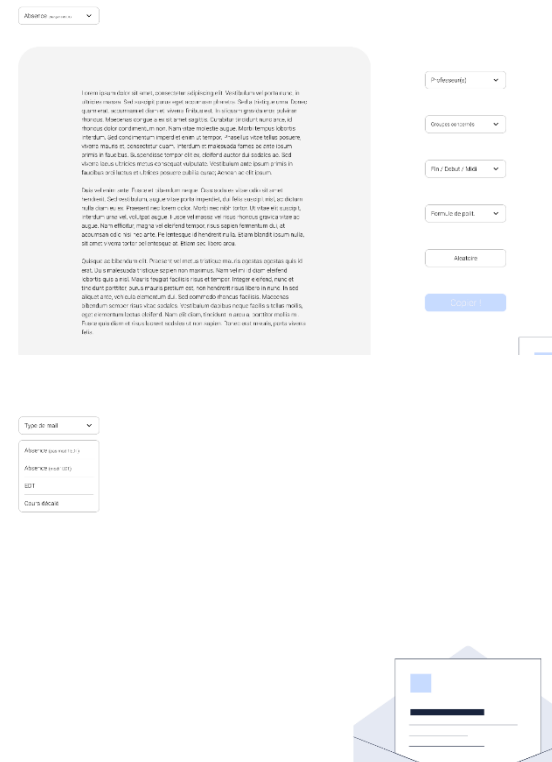
Maquette de la page d'accueil

”

Il faut que cette page d'accueil elle claque !

Alexandre Colas

L'homme aux idées douteuses



A ce stade du projet, le flow₅ de la page web était défini.

Il faut désormais passer à la **programmation** de la page en HTML pour la rendre **utilisable** et **concrète**.

2

Création des pages en HTML

Nous passons maintenant à du concret : réaction de la page en HTML.

Le squelette de page par défaut est utilisé pour notre page :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
</body>
</html>
```

Dans ce squelette nous ajoutons les premiers éléments de la page :

1. L'icône de fenêtre : le logo Saint-Adrien
2. Le titre : Vie Scolaire – Bienvenue
3. Des boutons menant aux pages de génération de mail et de gestion de base de données (future)
4. Le titre h1 (affiché sur la page) : Bienvenue dans l'utilitaire vie scolaire
5. Le sous-titre : Choisissez ci-dessous l'action désirée
6. L'image de fond : Un fondu de couleur de la collection hellomart

Le code de l'interface

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
2  <title>Vie scolaire - Bienvenue</title>
1  <link rel="icon" href="assets/img/Plan de travail 1Favicon.png">

</head>
<body>
  <div>
    <header>
      <div>
        <div>
          <div>
4             <h1>Bienvenue dans l'utilitaire vie scolaire</h1>
5             <div>
              <p>Choisissez ci-dessous l'action désirée</p>
            </div>
3             <a>
              Générateur de mails
              <i></i>
            </a>
3             <a>
              <strong>Gestion BDD</strong>
            </a>
          </div>
        </div>
      </div>
      <div>
        <div></div>
6        <div data-bg-image="assets/img/colorfades_lite_hellomart/1.jpg"></div>
      </div>
    </header>
    <main>
  </div>
</body>
</html>

```

Les formulaires et les liens entre pages

Le Formulaire :

Concentrons-nous un instant sur la création du formulaire qui est présent sur la page 'mail-generator.html' :

```
<form autocomplete="false">
  <h3>Concevons</h3>
  <p class="text-white d-none d-lg-block ts-opacity_50">Selectionner les differentes
informations du mail ci-dessous </p>
  <div>
    <input autocomplete="off" type="text" placeholder="Professeur" required>
  </div>
  <select>
    <option value="M. Tiedrez">M. Tiedrez</option>
    ...
  </select>
</div>
<button type="button">Copier</button>
<button type="reset">Reset</button>
</form>
```

Ceci est un formulaire HTML, il est contenu entre les bornes <form> et </form>. La borne ouvrante est celle qui admet les paramètres, elle a pour paramètre 'autocomplete=false' ce qui signifie que les popup de remplissage automatique des navigateurs n'apparaîtront pas. Le code du formulaire est ici volontairement simplifié pour ne voir qu'une fois chaque élément. Comme entrée du formulaire nous avons :

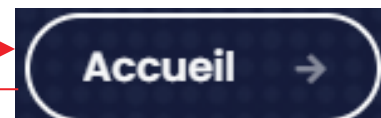
- <input name='nom du champ' autocomplete="off" required> : Cette entrée du form.6 est une entrée ouverte, elle garde comme paramètre l'autocomplétion7 désactivé, ainsi que la valeur 'required' qui signifie que le formulaire ne peut etre envoyé si ce champ n'est pas rempli.
- <select> <option value="M. Tiedrez">M. Tiedrez</option>... </select> : Cette entrée du formulaire est un choix multiple. Chaque choix contenu dans les balises <option> est paramétré par son argument value et le texte compris entre les valeur. La valeur de value est ce que le JS verra, et le texte entre les balises est le texte affiché dans la liste des choix.

Lien entre pages :



landing-page.html

```
<a href="mail-generator.html"
class="btn"> Générateur de mails </a>
```



mail-generator.html

```
<a href="landing-page.html"
class="btn"> Accueil </a>
```

Ajout des dépendances pour le style

La solution pour créer une page agréable d'utilisation et plaisante visuellement a été d'utiliser des frameworks et des dépendances

Bootstrap

Bibli. d'éléments

Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

Font Awsome

Icones polices

Font Awesome regroupe des polices d'écriture et un outil d'icônes qui se base sur CSS. Les éléments sont insérés grâce à la définition d'une class sur un div.

Animate

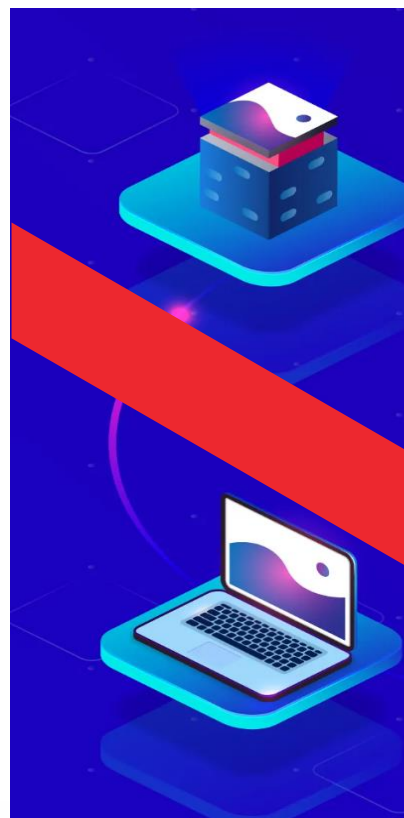
Animation de page

Animate.css est une bibliothèque d'animations multi-navigateurs prêtes à l'emploi à utiliser dans des projets Web. Idéal pour l'accentuation, les pages d'accueil, les curseurs et les conseils qui guident l'attention.

”

Framework (ang. m) :

Un *framework* est conçu en vue d'aider les programmeurs dans leur travail. C'est un ensemble d'outils et de composants logiciels formant un « squelette » de programme, un canevas.



```
3 * Copyright (c) 2013 Dmitrybaranov, 17
4 (function(e){var t,n,i,o,r,a,s,l="Close",c="BeforeClose",d="AfterClose",u="BeforeAppend",p="MarkupParse",f="Open",m="Change",
```

Pourquoi choisir l'utilisation de **frameworks**

R

Rapidité

Les frameworks sont hébergés sur des CDN :

content delivery network, ils sont constitués d'ordinateurs reliés en réseau qui coopèrent afin de mettre à disposition du contenu ou des données à des utilisateurs.

L'atout des CDN est leur vitesse car ils sont par définition imaginés pour offrir le routage le plus rapide vers les données.

F

Facilité d'utilisation

Créer une page web en utilisant un framework est un véritable gain de temps. Il suffit d'avoir la documentation à portée de main et d'un petit peu de créativité.

E

Esthétique

Un framework tel que Bootstrap permet d'avoir des éléments de page tous harmonieux qui, de plus, remplissent les critères d'accessibilité du web.

Comment intégrer les **frameworks**

4.1

Link

```
<link rel="stylesheet" href="assets/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" href="assets/font-awesome/css/fontawesome-all.min.css">
<link rel="stylesheet" href="assets/css/magnific-popup.css">
<link rel="stylesheet" href="assets/css/owl.carousel.min.css">
<link rel="stylesheet" href="assets/css/style.css">
```

Ces balises HTML sont insérées dans le <head> du document (l'en-tête head dans un document HTML est une partie du document qui n'est pas affichée par le navigateur au chargement de la page. Ce sont juste les métadonnées⁸ et les éléments à précharger) et permettent au document HTML de savoir vers où pointer pour charger les documents.

Ces documents sont enregistrés en local pour assurer une constante fonction du site. Ils sont enregistrés dans un format réduit dans lequel les sauts de ligne sont tous supprimés pour que l'entièreté du code ne forme qu'une ligne. Ceci dans le but d'accélérer le temps de chargement de la page.

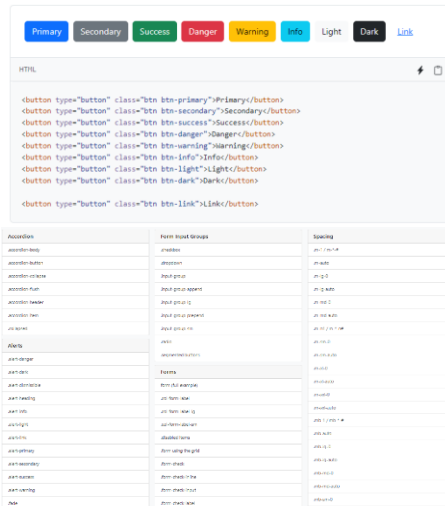
4.2

Attribution des Class

La plupart des frameworks fonctionnent de la même manière ; une documentation, constituée de noms de class à assigner aux éléments pour les styliser, est à utiliser en tant que compagnon pendant la programmation de la page. Voici quelques éléments de documentation des différentes dépendances :

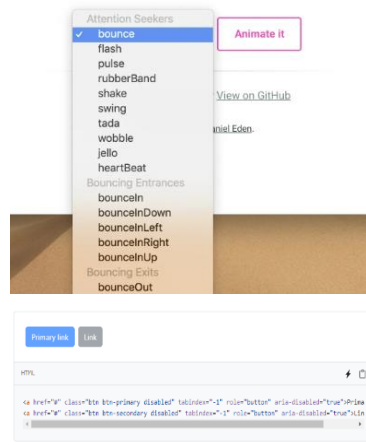
Exemples

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control.



Animate.css

Just-add-water CSS animations



Name	Class Name	Code	Example
Address Book	fa-address-book	f2b9	Try it
Address Card	fa-address-card	f2bb	Try it
Adjust	fa-adjust	f122	Try it
Air Freshener	fa-air-freshener	f5d0	Try it
Align Center	fa-align-center	f103	Try it
Align Justify	fa-align-justify	f109	Try it
Align Left	fa-align-left	f106	Try it
Align Right	fa-align-right	f108	Try it
Allergies	fa-allergies	f121	Try it
Ambulance	fa-ambulance	f099	Try it
American Sign Language Interpreting	fa-american-sign-language-interpreting	f2a3	Try it
Anchor	fa-anchor	f13d	Try it
Angle Double Down	fa-angle-double-down	f103	Try it
Angle Double Left	fa-angle-double-left	f100	Try it
Angle Double Right	fa-angle-double-right	f101	Try it
Angle Double Up	fa-angle-double-up	f102	Try it
Angle Down	fa-angle-down	f107	Try it
Angle Left	fa-angle-left	f104	Try it
Angle Right	fa-angle-right	f105	Try it

Dans notre HTML

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>Vie scolaire - Bienvenue</title>
  <link rel="icon" href="assets/img/Plan de travail 1Favicon.png">
</head>
<body>
  <div>
    <header>
      <div>
        <div>
          <div>
            <h1 data-animate="ts-fadeInUp">Bienvenue dans l'utilitaire vie scolaire</h1>
            <div>
              <p>Choisissez ci-dessous l'action désirée</p>
            </div>
            <a class="btn btn-primary btn-lg ts-scroll mr-4">
              Générateur de mails
            <i class="fa fa-arrow-right small ml-3 ts-opacity_50"></i>
            </a>
            <a>
              <strong>Gestion BDD</strong>
            </a>
          </div>
        </div>
      </div>
    </div>
    <div>
      <div></div>
      <div data-bg-image="assets/img/colorfades_lite_hellomart/1.jpg"></div>
    </div>
  </header>
  <main>
  </div>
</body>

```

Class de Animate

Class de Bootstrap

Class de font Awesome

Bootstrap et les layouts

Créer une page web responsive est un des plus grands challenges de la création de site internet. C'est un critère, car le web moderne est plus multiplateforme que jamais.

Qu'est-ce qu'une page responsive ?

Une page responsive est une page internet qui a la capacité



Qu'est ce qu'une page responsive ?

de s'adapter au support sur lequel elle est consultée, comme par exemple dans le cas d'un téléphone



Ce que propose Bootstrap



Un système de grille utilisant la méthode flexbox



Des tailles de colonnes et de lignes facilement définissables



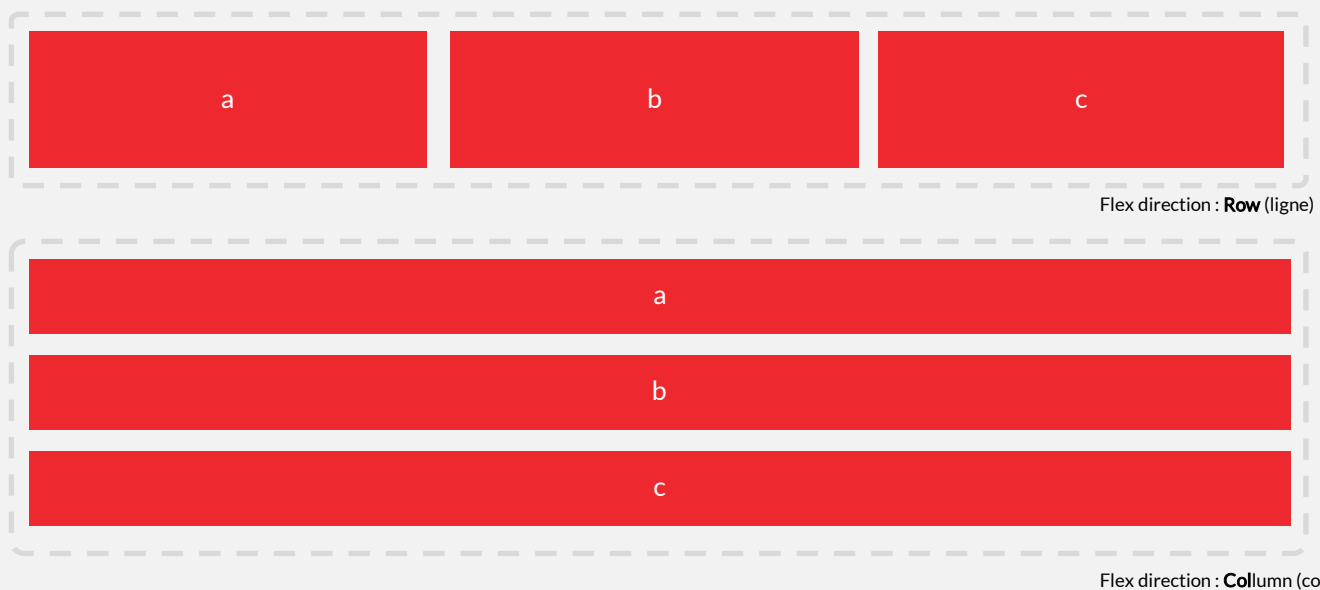
Des éléments qui s'adaptent à la taille de l'appareil grâce à l'utilisation de % dans le CSS

Exemple container responsif :

```
<div class="container align-self-center align-items-center">
  <div class="row">
    <div class="col-md-7">
```



- ➔ Les Flex Box CSS sont un standard CSS3 de disposition des éléments dans une page web. Ce standard permet d'avoir un design adaptatif à l'écran. Les éléments peuvent être réagencés selon la taille de l'écran.
- Class utilisées, **row** et **col-md-7**, définissent le tableau flexbox. C'est un tableau qui s'adaptera en orientation et en taille en gardant les mêmes proportions. Unique désavantage de flex est que le tableau est unidimensionnel.



Nous voilà arrivés à l'apparence **finale** de la page, nous devons désormais nous consacrer à la création **des fonctions** de l'outil pour le rendre **utilisable**

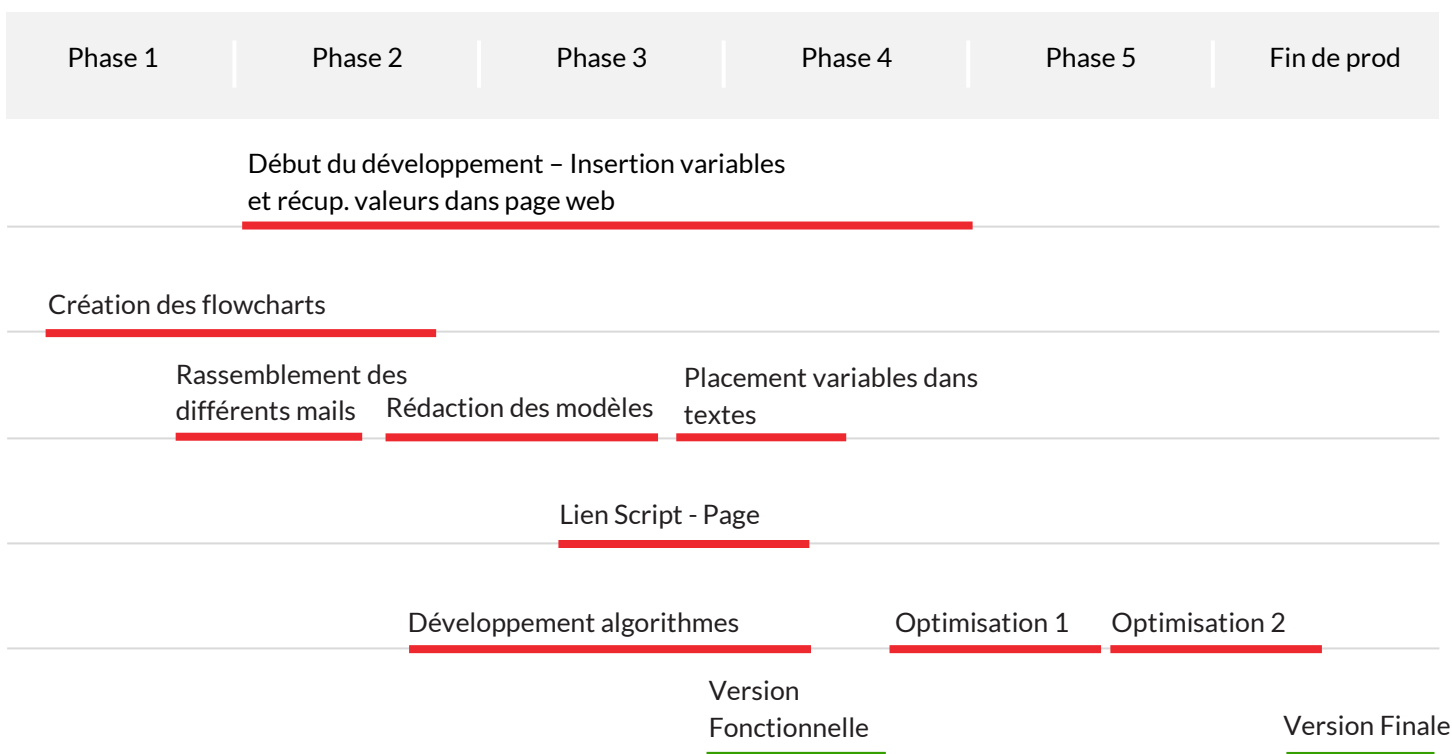
Voir annexe 4.4

Apparence finale des pages et flexbox documentation

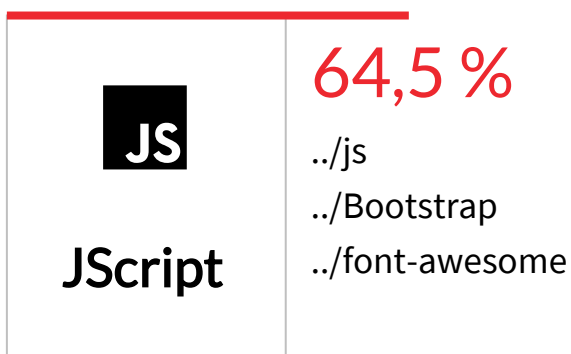
Ré – 2

Développement du code javascript

Le code JavaScript représente le plus grand défi de ce projet. C'était l'unique option abordable pour construire notre application web mais cela restait tout de même un nouveau langage de programmation. Avant d'entrer dans le détail, voici la chronologie du développement :



Proportion de technologies



64,5 %

../js
../Bootstrap
../font-awesome

Outre les fonctions de notre outil sur l'application web, plusieurs des dépendances (ré – 1) reposent sur des scripts JavaScript : principalement Bootstrap qui utilise JS pour plusieurs composants de son Framework. Contrairement à notre script, le code JavaScript de Bootstrap est dépendant de jQuery, c'est une bibliothèque JS permettant de faciliter le programme côté client pour des pages HTML.



Focus sur la création de l'application fonctionnelle

01

Qu'est-ce que JS Comment le manipuler

JavaScript est un langage de programmation de scripts principalement utilisé dans les pages web. Il est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web.

Lier le script à la page

```
<script src="assets/js/jquery-3.3.1.min.js"></script>
<script src="assets/bootstrap/js/bootstrap.min.js"></script>
<script src="assets/js/jquery-validate.bootstrap-tooltip.min.js"></script>
<script src="assets/js/custom.js"></script>
<script src="assets/js/script TEST ARRAY.js"></script>
```

Ces balises HTML sont insérées dans la fin du document, l'intérêt de les charger en fin de page est que si on mettait le JS tout en haut, la page n'aurait encore aucune donnée que l'affichage et le téléchargement seraient déjà bloqués : l'internaute n'aurait alors qu'une simple page blanche et rien à lire. Si le script est gros, cela peut prendre plusieurs secondes.

Ces fichiers JS sont tous locaux pour un chargement rapide et surtout pour permettre une maintenance facile.

02

Réinvestissement de connaissances en Python

Python et JavaScript présentent quelques similitudes notables en termes d'utilisation et de structure. Tous deux sont orientés objet, ce qui les rend idéaux pour le développement de logiciels complexes et de grande envergure. En outre, JavaScript et Python sont des langages de codage de haut niveau à typage dynamique, ce qui rend leur apprentissage assez facile.

Variables

Déclaration de variables :

Python :

```
textes = []
```

JavaScript :

```
textes = [] ;
```

Conditions et tests

Si, sinon, et égalités :

Python :

```
if type == 1 :
    texte_var = texte1_html
elif type == 2 :
    texte_var = texte2
```

JavaScript :

```
if (type.value == 1) {
    texte_var = texte1_html;
}
else if (type.value == 2) {
    texte_var = texte2;
}
```

Tableaux

Python :

```
textes = []
```

JavaScript :

```
textes = [] ;
```

Appel dans un tableau :

Python :

```
texte_var = textes[type-1]
```

JavaScript :

```
texte_var = textes[type.value-1];
```

Ces différents éléments de code sont les plus récurrents dans le programme lié à la page. Les similitudes avec python ont été d'une grande aide car elles ont permis d'accélérer le développement.

Python est le langage de programmation le plus étudié à l'école car il offre une simplicité d'utilisation, et une grande adaptabilité pour les erreurs de syntaxe ou de ponctuation du code. En classe de NSI, python est le sujet principal, abordé dès le début de l'année. Programmer en JavaScript présente donc un aspect de réinvestissement des compétences Python.

03

Base du développement

Maintenant que notre développeur a acquis les bases en JavaScript, il peut se lancer dans la rédaction du code. L'algorithme est basé sur le flowchart (page suivante) créé au préalable

Ce flowchart présente la manière dont la logique « s'écoule » dans notre programme. Le premier nœud du flowchart représente le sélecteur qui met en mémoire les *div* html où mettre le nouveau texte. On observe qu'au début plusieurs tests sont réalisés (certains ne figurent que sur le flowchart et pas le code final car ils représentent encore un trop gros défi) pour déterminer automatiquement des éléments basés sur les entrées de l'utilisateur. Ensuite une variable se voit assigner la valeur numérique associée au type de mail choisi : cette variable sera utilisée dans les crochets déterminant l'index du modèle de mail à sélectionner dans le tableau et à placer dans le *div*.

04

Le code commenté

Désormais, voyons point par point le code de l'application web. Nous allons découper le code en parties, qui seront commentées individuellement dans l'ordre dans lesquelles elles figurent dans le programme.

Ligne 1

```
const copiebutton = document.getElementById("copie-button");
```

Cette ligne contient la fonction qui attribue à la variable `copiebutton` un pointeur visant les éléments du document HTML portant l'id `#copie-button`. Dans la page 'mail-generator.html' cet élément est le bouton « copier » en bas du formulaire. Le mot clé `const` rend la variable `copiebutton` non réaffectable.

En JS, les variables peuvent être déclarées de plusieurs manières :

- `Var` : Simple déclaration, ce mot-clé doit être utilisé pour les vieux navigateurs
- `Const` : Déclaration non réaffectable
- `Let` : Déclare une variable 'localement', c'est-à-dire qu'elle ne portera sa valeur que dans la portée du bloc où elle a été annoncée
- `“ ” (vide)` : Même déclaration qu'en python

Ligne 3-10

```
textes = []
type = document.getElementById("mail-type");
prof = document.getElementById("fields1");
groupe = document.getElementById("fields2");
heure_deb = document.getElementById("fields3");
heure_fin = document.getElementById("fields4");
date = document.getElementById("fields-date");
cpe = document.getElementById("fields-user");
```

La première ligne n'est qu'une initialisation de variable

Les autres lignes ont pour but de récupérer dans des variables les valeurs renseignées par l'utilisateur sur la page HTML. La fonction utilisée est la même que pour le sélecteur de bouton en première ligne :

`document.getElementById("id")` cette fonction sélectionne les éléments portant l'id visé car ce sont des éléments de formulaire. Les valeurs y étant entrées seront accessibles dans le script grâce à la fonction `'.value'` qui sera par exemple utilisée ainsi : `prof.value`

Ligne 13-80

```
copiebutton.addEventListener("click", () => {
    var fieldNameElement = document.getElementById('replace');
    date = date.value;
    date_m = date.toString().substring(5,7);
    date_d = date.toString().substring(8,10);
    date = `${date_d}/${date_m}`
    textes = [...];
```

La première ligne est une fonction qui pause le programme en l'attente d'un 'event', ici l'évènement est défini pour être de type "click" sur l'élément `copiebutton`. La syntaxe de cette fonction est la suivante :

`element.addEventListener("click", myFunction);`

Dans un souci d'efficacité, la fonction exécutée par l'eventlistener¹⁰ est directement rédigée dans les parenthèses des paramètres de la fonction.

La ligne suivante est celle pointant vers l'objet `div` qui contiendra le mail généré.

Les 4 prochaines lignes servent à reformater la date, elle est originellement au format AAAA-MM-JJ, un format auquel les Français ne sont pas habitués. Le première ligne met en variable la valeur de la date rentrée, la deuxième et troisième lignes utilisent les fonctions `.toString()` et `.substring()`. La première fonction convertit la variable numérique en texte, et la deuxième fonction supprime une portion du texte grâce aux indices mis en paramètre. Enfin, la quatrième ligne est une concaténation entre la variable contenant le mois isolé, et le jour isolé.



La dernière ligne est le tableau 'textes' contenant tous les modèles de mail.

Un des textes

```
`  
Madame, Monsieur,  
<br>  
<br>  
Suite à l'absence de ${prof.value} ce ${date}, les élèves externes du groupe  
${groupe.value} commenceront les cours de l'après-midi à ${heure_deb.value} ce  
${date}.  
<br>  
<br>  
Vous souhaitant bonne réception,  
<br>  
Cordialement,  
<br>  
${cpe.value}  
<br>  
Réfèrent Vie Scolaire  
`
```

Voici un des modèles de mail, dans chacun des modèles, les informations clés faisant partie de notre formulaire sont remplacées par un objet ayant cette syntaxe : `${variable.value}` cet élément sert à effectuer une concaténation¹¹ entre le texte pré écrit et une variable sans avoir à additionner plusieurs strings entre eux. Cette méthode ne marche que dans des strings contenus entre deux accents graves.

Une irrégularité se note dans la syntaxe de la concaténation de `${date}` car, comme nous l'avons vu dans la portion de code au-dessus, la variable `date.value` a été découpée puis réinsérée dans la variable `date`.

On note aussi l'utilisation des balises `
` qui affichent un saut de ligne sur la page quand le navigateur fait le rendu de la page

En outre, comme en python, les différents éléments d'un tableau sont séparés par une virgule en JavaScript.

Ligne 81-Fin

```
texte_var = textes[type.value-1];  
navigator.clipboard  
  .writeText(texte_var.replace(/<br>/g, ''))  
  .then(() => {  
    console.log("Text copied to clipboard...");  
  })  
  .catch((err) => {  
    console.log("Something went wrong", err);  
    fieldNameElement.innerHTML = texte_var;  
  })
```

Cette portion finale représente le remplacement du texte dans le document ainsi que la copie automatique du texte généré dans le presse papier.

La première ligne définit la variable `texte_var` dans laquelle vient être placé l'élément du tableau 'textes' d'index `type.value-1`. La partie `type.value` représente la valeur numérique associée au premier sélecteur du formulaire, et la soustraction de 1 sert à adapter la valeur au système d'index de JS : partant de 0 pour la première valeur.

La deuxième ligne appelle la fonction `navigator.clipboard` à laquelle l'argument `.writeText()` est associé. Cet argument à la troisième ligne prend pour paramètre l'expression suivante :

`texte_var.replace(/
/g, ' ')` cette expression prend le texte généré et effectue la fonction `replace` dessus, avec en premier paramètre `/
/g` et en deuxième `' '`. Le premier est le remplacé (g signifie toutes les occurrences) et le deuxième le remplaçant qui ici, est un espace.

En résumé, ces deux lignes ont pour but de copier le texte dans le presse papier, mais privé de ses balises HTML de saut de ligne, qui auraient simplement été interprétées comme du texte dans école directe.

Les 5 lignes suivantes ne sont qu'à vocation de débogage, elles indiquent dans la console si le texte a été copié ou non.

Finalement, la dernière ligne insère la variable 'texte_var' dans le document HTML avec la fonction `fieldNameElement.innerHTML` la partie avant le point de cette fonction est une variable pointant vers les éléments portant l'id replace. La variable 'texte_var' comporte toujours les balises de saut de ligne car les moteurs de rendu des navigateurs ne prennent pas en compte des simples sauts de ligne dans un document HTML

05

Mise en relation interface - variables

Un peu perdu après toutes ces explications de code ? On va se poser avec un petit café et mettre en lien notre interface et notre code fraîchement expliqué.

Concevons

Sélectionner les différentes informations du mail ci-dessous

Type de mail

Professeur

Groupes concernés

jj/mm/aaaa

Heure début

Heure fin

Utilisateur

Copier

Reset

Variables associées

```
type = document.getElementById("mail-type");  
- Valeur numérique de 1 à 4 / Requis  
prof = document.getElementById("fields1");  
- String libre / Requis  
groupe = document.getElementById("fields2");  
- String libre / Requis  
heure_deb = document.getElementById("fields3");  
- Date format AAAA-MM-JJ / Requis  
heure_fin = document.getElementById("fields4");  
- String non libre (10h05, 11h05, ...)  
date = document.getElementById("fields-date");  
- String non libre (14h25, 15h25, ...)  
cpe = document.getElementById("fields-user");  
- String non libre (M. Tiedrez) / Requis
```

Premier Jet

```

copiebutton.addEventListener("click", () => {
  if (type.value == 1) {
    navigator.clipboard
      .writeText(texte1) // copier dans le presse papier
      .then(() => {
        console.log("Text copied to clipboard...");
      })
      .catch((err) => {
        console.log("Something went wrong", err);
      });
    //selectionner la cible du remplacement
    var fieldNameElement = document.getElementById('replace');
    //remplace contenu du div ciblé
    fieldNameElement.innerHTML = ` ${texte1} `;
  }
  else if (type.value == 2) {
    navigator.clipboard
      .writeText(texte2) // copier dans le presse papier
      .then(() => {
        console.log("Text copied to clipboard...");
      })
      .catch((err) => {
        console.log("Something went wrong", err);
      });
    //selectionner la cible du remplacement
    var fieldNameElement = document.getElementById('replace');
    //remplace contenu du div ciblé
    fieldNameElement.innerHTML = ` ${texte2} `;
  } ...
}

```

Cette première version utilisait une suite de tests pour sélectionner le bon modèle à appliquer. Non seulement le coût en temps de cet algorithme était très sous optimal, mais le nombre de tests est drastiquement supérieur au nombre de tests de la version finale. Cette version utilise 17 variables et le code de sélection fait 100 lignes.

Optimisation 1

```

copiebutton.addEventListener("click", () => {
  var fieldNameElement = document.getElementById('replace');
  date = date.value;
  date = date.toString().substring(5);
  //Tout les types de mails
  texte1_html = ``;
  ...
  texte_var = '';
  if (type.value == 1) {

```



```

        texte_var = texte1_html
    }
    else if (type.value == 2) {
        texte_var = texte2
    }
    ...
    navigator.clipboard
        .writeText(texte_var.replace(/<br>/g, '')) // copier dans le presse
        papier en supprimant tt les occurrences de <br>
        .then(() => {
            console.log("Text copied to clipboard...");
        })
        .catch((err) => {
            console.log("Something went wrong", err);
        });
    //selectionner la cible du remplacement
    //remplace contenu du div ciblé
    fieldNameElement.innerHTML = ` ${texte_var} `;

});

```

Cette deuxième version introduit la méthode pour supprimer les balises HTML du texte copié, ce qui divise par deux le nombre de variables de textes. Des tests subsistent pour sélectionner le bon modèle à appliquer. Cette première optimisation diminue le nombre de variables en les faisant passer de 17 à 13. Le nombre de tests ne diminue pas, mais le nombre de lignes passe de 100 à 32. Le coût en temps est divisé par trois en n'exécutant qu'une fois les fonctions qui ne dépendent pas du type de mail choisi.

Dernière version

```

copiebutton.addEventListener("click", () => {
    var fieldNameElement = document.getElementById('replace');
    date = date.value;
    date_m = date.toString().substring(5,7);
    date_d = date.toString().substring(8,10);
    date = `${date_d}/${date_m}`;
    textes = [ `` ];
    texte_var = textes[type.value-1];
    navigator.clipboard
        .writeText(texte_var.replace(/<br>/g, ''))
        .then(() => {
            console.log("Text copied to clipboard...");
        })
        .catch((err) => {
            console.log("Something went wrong", err);
        });
    fieldNameElement.innerHTML = texte_var;
});

```

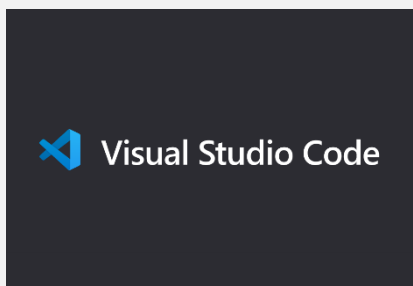
Cette dernière version introduit l'utilisation d'un tableau pour les textes, -4 en nombre de variables, de plus aucun test n'est effectué dans cette nouvelle version. Il est remplacé par l'utilisation de la valeur numérique associée au type de mail choisi en tant qu'index pour sélectionner le texte dans le tableau.

Au final, le nombre de lignes est à nouveau réduit : passant de 32 à 17. De plus 1 variable est utilisée au lieu de 4 pour les textes. Cette version est totalement compatible en cas d'ajout de nouveaux textes.

A notre disposition

De nombreux outils

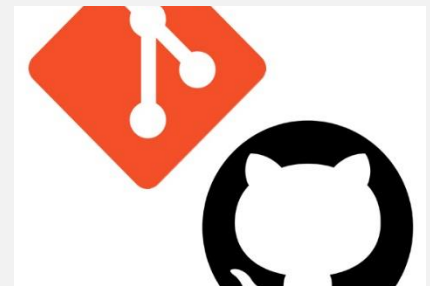
Pour développer notre solution, nous nous sommes aidés de plusieurs outils **indispensables**. Sous forme de **site internet**, ou bien de **logiciels** ces outils ont été primordiaux



Visual Studio Code

Visual Studio Code est un éditeur de code extensible développé par Microsoft. Les avantages de cet IDE^{12a} sont la prise en charge du débogage avec breakpoints¹³ faciles à mettre en place, la mise en évidence de la syntaxe, la complétion intelligente du code avec les extension développées par Microsoft telles que IntelliSense, ainsi que la refactorisation du code et le reformatage automatique, ou bien Git intégré pour du développement distant.

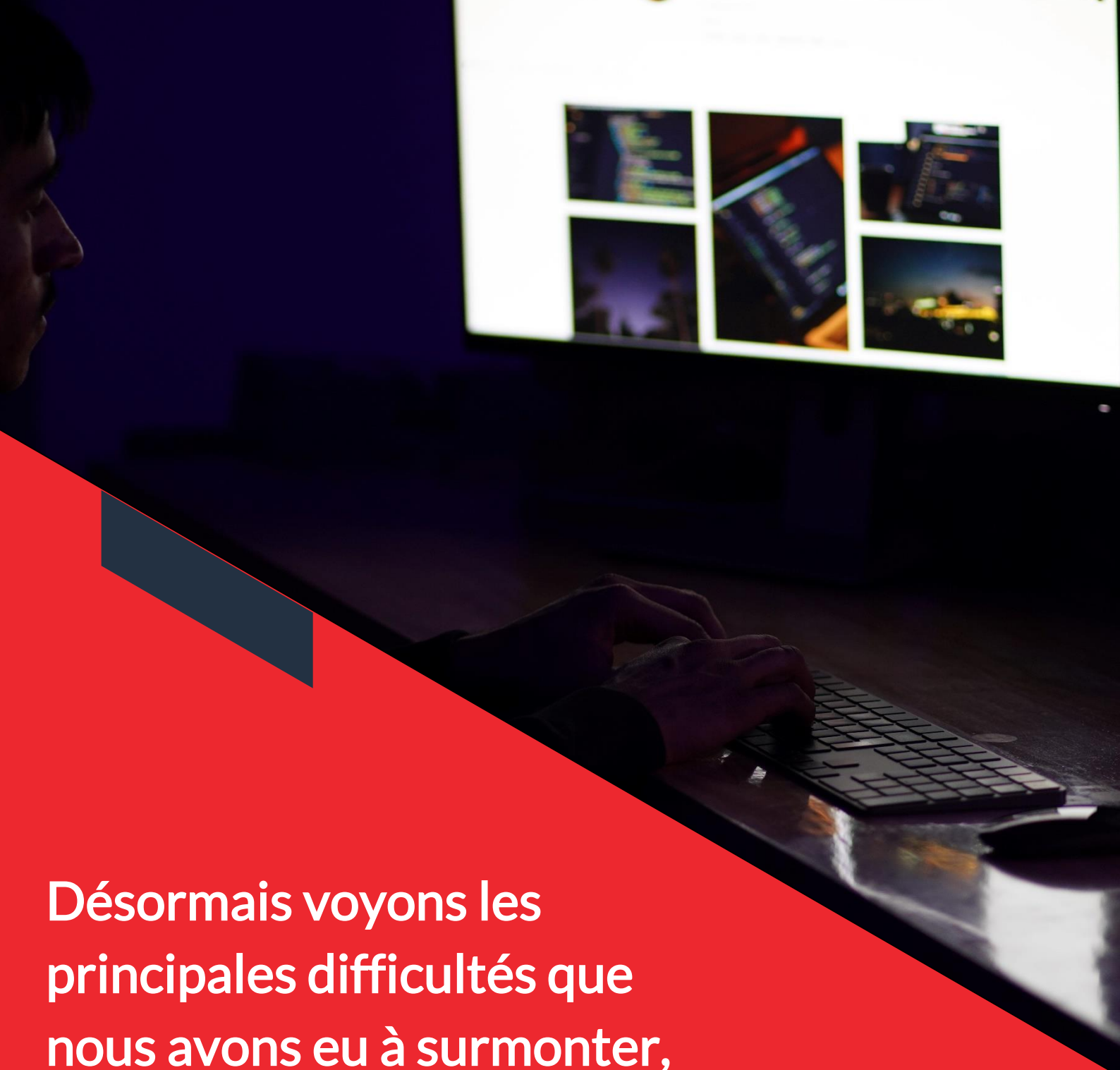
Le plus grand atout de cet outil dans la création du projet a été l'extensibilité, tout au long du développement le codeur a utilisé l'extension 'Live Server' qui permet de programmer une page web de de voir le résultat en direct au moindre changement du fichier HTML, CSS ou bien JS



GitHub et Git

Git est un outil de contrôle de source, c'est à dire c'est une solution permettant de garder une chronologie des version d'un fichier tout au long de sa modification. Git est un service largement utilisé dans le monde du développement car il facilite grandement la tâche de garder des versions antérieures. Cette fonction s'est vue utile lors des phases d'optimisation car quelques erreurs irréversibles avaient été commises.

GitHub lui est une plateforme de partage de code, elle se base sur git pour le contrôle de source. Plusieurs fonctionnalités de GitHub ont été essentielles dans le projet. Notamment l'hébergement de code accessible via quelconque PC doté de VS code, cette fonction m'a permis de travailler sur l'outil durant des voyages s'étant déroulés pendant les phases de développement. Autre fonctionnalité importante : l'hébergement de page web gratuitement, nous y reviendront en détail dans la section suivante.



Désormais voyons les principales difficultés que nous avons eu à surmonter, et certaines que nous n'avons pas réussi à franchir.

« Parfois lorsque vous innovez, vous faites des erreurs. Il est préférable de les admettre rapidement et de continuer à améliorer vos autres innovations. »

Steve Jobs

Obstacle

+ Le JavaScript

- Créer des Listes
 - Récupérer texte dans formulaire
 - Insérer variables dans String
 - Accéder au presse papier
 - Savoir quand un bouton est pressé
-
- Reformatage de la date

+ Page web

- Créer une page web responsive
- Bug de 'seconde chance'

+ Intégration

- Lien école directe

Solution

Pour trouver des solutions au problème que le JS posait, nous nous sommes appuyés sur nos connaissances en python, des recherches internet et la précieuse aide de Mathéo Dupont. Il a aidé le développeur sur la question du presse papier ainsi que de la récupération de valeurs dans le formulaire.

N'ayant trouvé aucune fonction permettant un reformatage de la date facile, le développeur a opté pour un contournement du problème détaillé à la page 27

La page devait être adapté et utilisable sur n'importe quel ordinateur que le CPE serait amené à utiliser, la solution a été d'utiliser les grilles flexbox de Bootstrap. Ensuite, le bug cité est un problème qui survient lorsque que l'on essaye de générer un nouveau mail avec les champs préremplis d'une génération précédente. Dans ce cas de figure le bouton copier ne mène à rien et il faut rafraichir la page

Suite à plusieurs échanges avec Mme Dollé, la responsable informatique de Saint-Adrien, nous avons compris qu'il n'était pas possible de poser un lien dans la barre latérale d'école directe. La solution trouvée sera détaillée dans la section dédiée (5)

Conclusion

Toute cette période de production a été **enrichissante**, aussi bien **techniquement** qu'**humainement**. La collaboration a pu se voir **difficile** au sein de l'équipe, mais elle aura fini par se faire, et ainsi nous amener à la section 5, le déploiement de l'outil fini.

Les obstacles ont été **omniprésents** tout au long du projet, mais que serait un projet sans obstacle ! Malheureusement, certaines difficultés nous ont contraint de réduire certaines fonctionnalités de l'outil, mais le ressortissant est un produit **fini, stable et soigné**.

Le Déploiement

Le déploiement se passe en deux étapes : **héberger** l'outil, puis **l'intégrer**.

Date de
mise en
ligne

07 Mai 2022

Point de départ Interface seulement

Hébergement

Héberger signifie, dans le monde de l'informatique et d'Internet, le fait de mettre à disposition des créateurs de sites Web des espaces de stockage sur des serveurs sécurisés, afin que les sites Web en question puissent être accessibles sur le **www**. En résumé il s'agit d'un service permettant de placer, sur un serveur relié au réseau Internet, les différents fichiers et contenus qui constituent un site Web.

Deux solutions d'hébergement s'offrent à nous, toutes deux aussi bien adaptées. La première est l'hébergement local qui consiste à placer le site web dans son entièreté sur un serveur de stockage local (Ex : serveur Quartz de Saint-Adrien). Le site devient accessible à tout utilisateur autorisé étant sur le même réseau que les fichiers via l'adresse locale du fichier (Ex : 198.168.1.34/mail-tool/landing-page.html)

La seconde option est l'hébergement en ligne, sur le **www**. Cet hébergement offre un accès aux fichiers (donc au site) à quelconque appareil connecté à internet via une URL publique.



+ Du local

- Ne dépend pas de la connexion à internet.
- Rapidité de chargement.

+ Sur internet

- Adresse sous forme d'URL.
- Maintenance possible à distance.
- Stabilité et sécurité des fichiers.

- Du local

- Adresse pas sous forme d'url.
- Maintenance possible qu'en local.
- Gérer les accès aux fichiers.

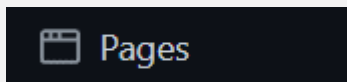
- Sur internet

- Nécessite la connexion au web
- Chargement 0,2 seconde plus lent
- Dépend de l'hébergeur.

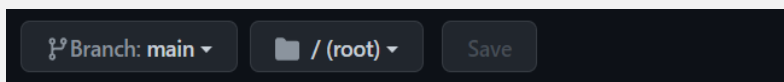
➔ Notre choix

La solution d'hébergement pour laquelle nous avons opté est l'hébergement en ligne, grâce au service 'pages' de GitHub. Voici comment déployer son site web sur GitHub :

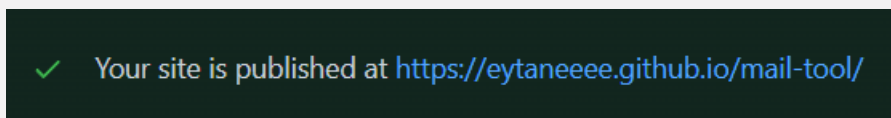
1. Une fois le 'repository' publié sur github, il faut se rendre dans les paramètres du répertoire et se rendre dans la section 'pages'



2. Choisir la source (la plupart du temps il faut choisir la branche principale et la racine du répertoire)



3. **Et voilà !** Le site est accessible à l'url au format : *utilisateur.github.io/nom-de-répertoire/nom-de-fichier*



Implémentation

La toute dernière étape significative du projet est l'implémentation de l'outil dans Saint-Adrien, et plus précisément dans le workflow des intéressés : les CPE.

Cette implémentation prend pour point de départ la mise à disposition de l'outil. L'idée originale pour mettre l'outil à disposition était un lien dans la barre latérale école directe au même niveau que les liens ci-contre.

Malheureusement cette solution s'est avérée impossible après un bref entretien avec Mme Dollé. Elle nous a informés que les liens présents dans cette barre latérale étaient prédéfinis par l'éditeur de école directe et n'étaient qu'activés par ses soins, et non choisis. Cet obstacle a été gênant car il s'est présenté au dernier moment alors que nous imaginions que cette solution aurait été possible. La réponse à cet accroc sera de placer le lien sur la page d'accueil du personnel en 'post-it' comme il vous sera démontré lors du test technique.

Ce qui suit la mise à disposition est la mise en usage, c'est-à-dire de 'former' le personnel intéressé pour lui présenter l'outil et lui donner toutes les clés en main pour, s'il le souhaite, qu'il intègre la solution à son travail.

Retours :

Malheureusement, compte tenu des dates et du déroulement de cette fin d'année, l'outil n'a pas encore pu être présenté à l'équipe de CPE, nous n'avons donc pas pu recueillir de retours. Nous espérons pouvoir avoir un moment avec l'équipe de CPE et les professeurs de NSI pour présenter la solution et recueillir des réactions.



Vers l'infini et au-delà

Le futur / Conclusion

Nous voilà, après **4 mois** de travail à la fin de ce rapport, mais aussi de ce projet !
Tant de choses ont été **dites**, **expliquées**, **imaginées**, **rêvées**. Cette section va résumer quelques idées et pistes d'**améliorations** qui sont encore **désirées** par l'équipe

Interface

Améliorations possibles

Meilleure responsivité pour l'utilisation sur mobile. Réglage du bug de 'seconde chance'. Ouverture automatique du site lors de l'accès à la page de rédaction de mail sur E.D. Génération de mails en live. Rassemblement en un des deux sélecteurs d'horaire

Logique

Améliorations envisagées

Plus large prise en charge de type de mails : ajout de tests complexes et de concaténations encore plus poussées pour donner un aspect d'intelligence à l'outil.

Implémentation

Améliorations rêvées

Un objectif ultime serait d'intégrer l'outil directement dans école directe, lors du clic sur le bouton 'nouveau message' depuis le compte d'un CPE. A moins que notre outil devienne mondialement connu, cette idée à peu de chances de se réaliser.

Autre amélioration serait de pouvoir mettre le lien dans la barre de tache école directe du personnel, pour cela il faudrait que l'outil s'avère réellement intégré dans le workflow des CPE

Bases de données

Ajout envisagé

Un vrai + à l'outil serait l'ajout de bases de données pour gérer les messages pré écrits, les horaires du sélecteur, une liste des noms de profs, de groupes etc... Pour, premièrement, permettre à l'outil d'avoir une durée de vie élevé dans St-Adrien, mais aussi deuxièmement, de le rendre accessible a n'importe quel établissement scolaire.

Cette fonctionnalité reposerait certainement sur un fichier CSV formatté, et sur un programme python ou javascript étant capable de l'interpréter. Le lien vers une base de données est déjà prévu sur la page (mais ne mène a rien) et la maquette de cette page avait été réalisée :

Choisissez ci-dessous l'action désirée

Générateur de mails →

Gestion BDD

Gestion BDD

La gestion des bases de données se présentera sous la forme d'un menu de paramètres avec tous les éléments paramétrables

Gestion BDD

Professeurs

Classes

Matières

Modèles

Gestion BDD

Professeurs

Classes

Matières

Modèles

Absence 1

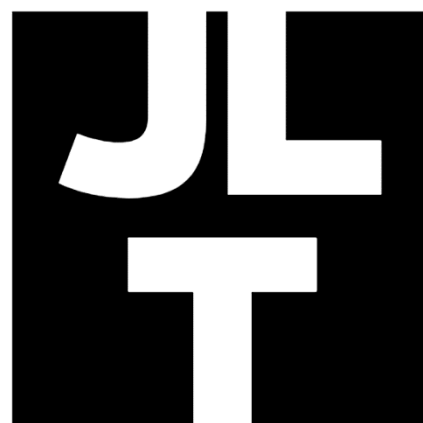
Absence 2

Amenagement EDT

Journée pedago

Absence 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum vel porta nunc, in ultricies massa. Sed suscipit purus eget accumsan pharetra. Sed a tristique urna. Donec quam erat, accumsan et diam et, viverra finibus est. In aliquam gravida eros pulvinar rhoncus. Maecenas congue a ex sit amet sagittis. Curabitur tincidunt nunc ante, id rhoncus dolor condimentum non. Nam vitae molestie augue. Morbi tempus lobortis interdum. Sed condimentum imperdiet enim ut tempor. Phasellus vitae tellus posuere, viverra mauris et, consectetur quam. Interdum et malesuada fames ac ante ipsum primis in faucibus.



JLT

Jamai'l le temps software
15 Rue Jean-Baptiste de la Salle,
59650 Villeneuve-d'Ascq

E : jlt@coolsite.net

eytaneeee.github.io/mail-tool/lansing-page.html

Un très grand
merci pour votre
intérêt porté à
notre projet