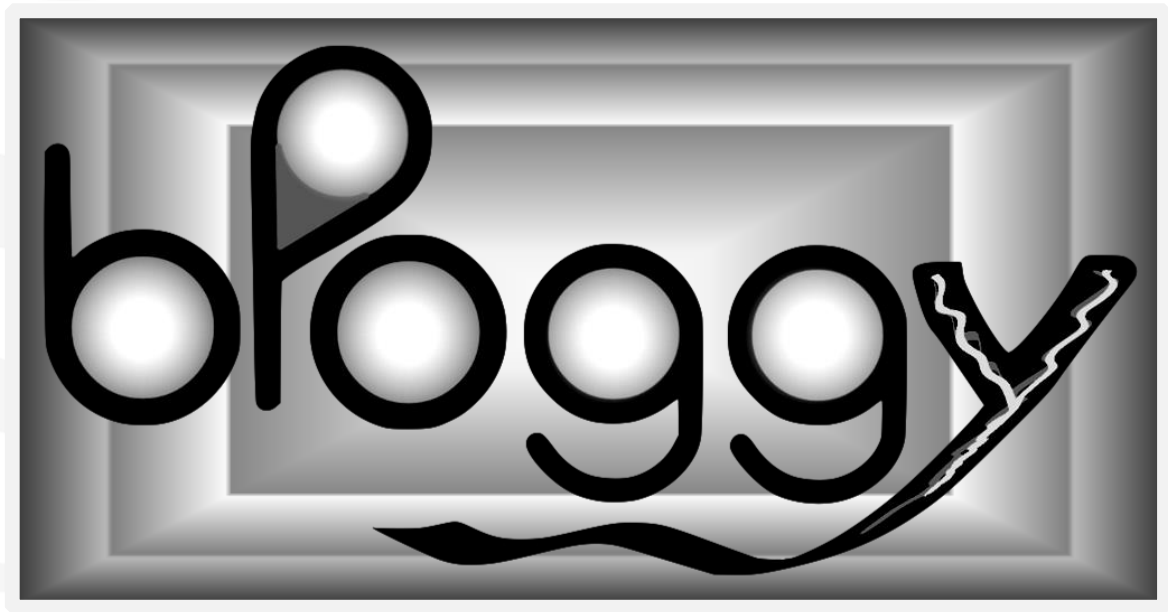


# RAPPORT DE PROJET - NSI

## " BLOGGY "



**Projet:** Site web de gestion de base de donnée type blog

**Réalisé par:** Mélanie BULKAN

(classe de première)

# **SOMMAIRE**

<b>SOMMAIRE .....</b>	<b>- 2 -</b>
<b>INTRODUCTION.....</b>	<b>- 3 -</b>
<b>DESCRIPTION DU PROJET.....</b>	<b>- 4 -</b>
<b>A. Production minimale viable.....</b>	<b>- 4 -</b>
<b>B.Version intermédiaire .....</b>	<b>- 6 -</b>
<b>C.Version finale (validation du projet) .....</b>	<b>- 7 -</b>
<b>INTERET DU PROJET.....</b>	<b>- 11 -</b>
<b>A. Pourquoi avoir choisi ce projet ?.....</b>	<b>- 11 -</b>
<b>B. Quels intérêts apporte Bloggy ?.....</b>	<b>- 11 -</b>
<b>SUIVI DU PROJET.....</b>	<b>- 12 -</b>
<b>A. Planning prévisionnel.....</b>	<b>- 12 -</b>
<b>B.Gestion des aléas.....</b>	<b>- 13 -</b>
<b>C.Temps consacré à la réalisation .....</b>	<b>- 13 -</b>
<b>PROGRAMMATION DU PROJET .....</b>	<b>- 15 -</b>
<b>A. Organisation des pages web.....</b>	<b>- 15 -</b>
<b>B.Structure de la base de donnée.....</b>	<b>- 16 -</b>
<b>C.Logiciels et langages utilisés.....</b>	<b>- 17 -</b>
<b>D. Les contraintes et leurs solutions.....</b>	<b>- 17 -</b>
<b>ANNEXES .....</b>	<b>- 19 -</b>
<b>A. Sources utilisés pour l'apprentissage .....</b>	<b>- 19 -</b>
<b>B.Autres sources .....</b>	<b>- 19 -</b>
<b>CONCLUSION.....</b>	<b>- 20 -</b>

# INTRODUCTION

D'après le programme officiel, un enseignement d'informatique ne saurait se réduire à une présentation de concepts ou de méthodes sans permettre aux élèves de se les approprier en développant des projets.

Un tiers au moins de l'horaire total de la spécialité est réservé à la conception et à l'élaboration de projets conduits par des petits groupes d'élèves. Les projets réalisés par les élèves, sous la conduite du professeur, constituent un apprentissage fondamental tant pour l'appropriation des concepts informatiques que pour l'acquisition de compétences.

En classe de première comme en classe terminale, ils peuvent porter sur des thèmes variées ou des problématiques issues d'autres disciplines. Ces projets peuvent êtres de contenu ou de formes diverses et ont essentiellement pour but d'imaginer des solutions répondant à un problème.

Ainsi, étant élève de première (suivant la spécialité NSI) il m'était proposé de réaliser un projet me permettant de mettre en pratique mes connaissances acquises au cours de cette année. Ma grande passion pour cette matière, surtout dans le domaine de la création de pages web m'a orienté vers un projet ambitieux de création de site avec gestion de base de données nonobstant le fait que les connaissances requises pour la réalisation de ce projet n'étaient pas du niveau de première.

Ce rapport à pour but d'analyser les étapes clés de la conception de ce projet. En passant par l'organisation de l'emploi du temps, du travail, à la conception, réalisation et aux aléas que chacun à du faire face. Nous allons donc retracer le parcours qui nous à mené à notre version finale de projet

# DESCRIPTION DU PROJET

## A. Production minimale viable

La base de mon projet consiste en la création d'un site web avec gestion de base de données où plusieurs utilisateurs peuvent s'échanger des informations. De ce fait, la version minimale (MVP) se doit de répondre à certaines conditions :

Créer un site opérationnel avec peu ou pas de style (sans CSS).

Proposer au client de remplir un formulaire avec une case "pseudo" et une autre case "message".

Puis, lorsqu'il appuie sur "envoyer" le contenu de son message apparaît sur le site, ainsi que la date à laquelle il l'a envoyé et son pseudo.

Donc le site affiche les 20 derniers messages qui ont été envoyés ainsi que quelques informations complémentaires.

Voici un exemple de code pour répondre à ces besoins en deux pages :

*Minichat.php : pour l'affichage de la page (contient le corps html)*

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8" />
5          <title>tchat</title>
6      </head>
7      <h1> SUPER TCHAT ONLINE</h1>
8      <p>Veuillez entrer votre pseudo et message pour l'envoyer dans le tchat :)</p>
9      <form action="minichat2.php" method="post">
10         <p>
11             <label for="pseudo">Votre pseudo :</label>
12             <input type="text" name="pseudo" id="pseudo" /><br />
13             <label for="message">Votre message :</label>
14             <input type="text" name="message" id="message" /><br />
15             <input type="submit" value="Envoyer" />
16         </p>
17     </form>
18 <body>
19 <?php
20 try
21 {
22     $bdd = new PDO('mysql:host=localhost;dbname=test;charset=utf8', 'root', 'root'
23 );
24 }
25 catch(Exception $e) // Pour afficher les erreurs
26 {
27     die('Erreur : '.$e->getMessage());
28 }
29
30 $reponse = $bdd->query('SELECT pseudo, message, datee FROM minichat ORDER BY ID
31 DESC LIMIT 0,10');
32
33 while ($donnees=$reponse->fetch())
34 {
35     echo '<p><strong>'.htmlspecialchars($donnees['pseudo']).'</strong> : '.
36         htmlspecialchars($donnees['message']). '<br /><i>le :'. $donnees['datee']
37         .'</i></p>';
38 }
39
40 $reponse->closeCursor();
41
42 ?>
43 </body>
44 </html>

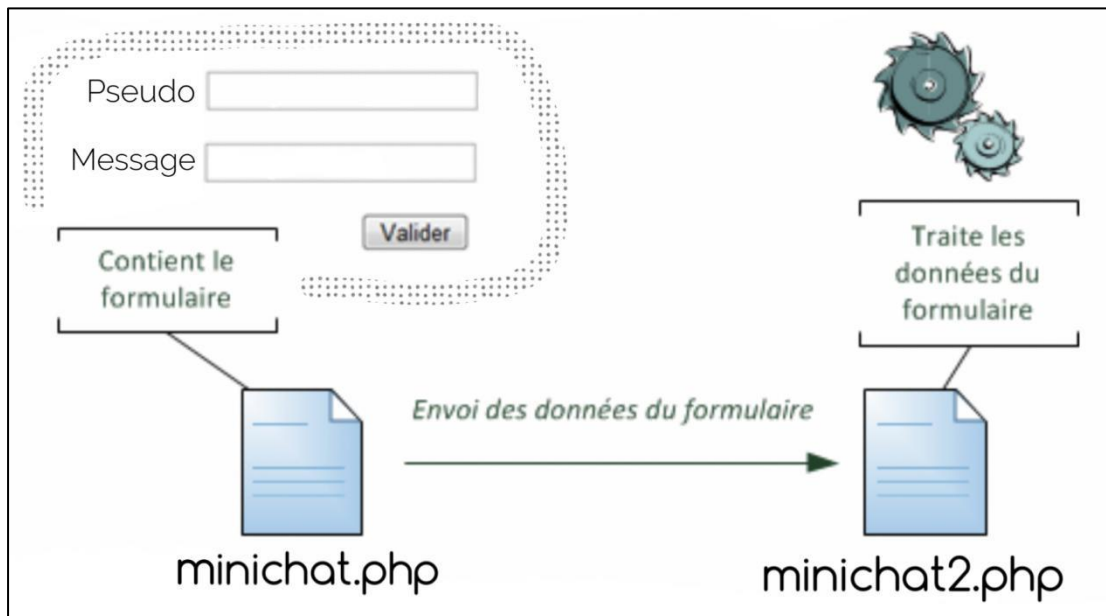
```

*minichat2.php : pour le traitement (php) des données du formulaire*

```

1  <?php
2
3
4      // Connexion à la base de données
5      try
6      {
7          $bdd = new PDO('mysql:host=localhost;dbname=test;charset=utf8', 'root', 'root'
8 );
9      }
10     catch(Exception $e)
11     {
12         die('Erreur : '.$e->getMessage());
13     }
14     // Insertion du message à l'aide d'une requête préparée
15     $req = $bdd->prepare('INSERT INTO minichat (pseudo, message) VALUES(?, ?)');
16     $req->execute(array($_POST['pseudo'], $_POST['message']));
17
18     header('Location: /minichat/minichat.php');
19
20 ?>

```



Bien sur, il a aussi fallu créer une table SQL “minichat” abritant plusieurs champs :

- “id” (clé primaire en auto-incrément)
- “pseudo” (pseudo entré dans le formulaire de la page minichat.php)
- “message” (message entré dans le formulaire de la page minichat.php)
- “datee” (qui contient la date à laquelle à été envoyé le message)

## B. Version intermédiaire

La version minimale est loin de répondre à toutes mes attentes. En effet, elle se contente uniquement d’afficher n’importe quelle information que lui envoie n’importe quel visiteur. Elle présente de grand soucis de sécurité (failles), d’identité (des utilisateurs qui peuvent prendre n’importe quel pseudo). De plus, le projet n’est pas esthétique et propose peu de fonctionnalités. Pour remédier à cela:

Ajouter une page CSS pour styliser la page.

Proposer un système d’inscription/connexion plus sécurisé avec un identifiant et un mot de passe. Comme ça, chaque utilisateur a son propre pseudo ce qui règle les problèmes d’identités

Créer un menu de navigation avec plusieurs rubriques:

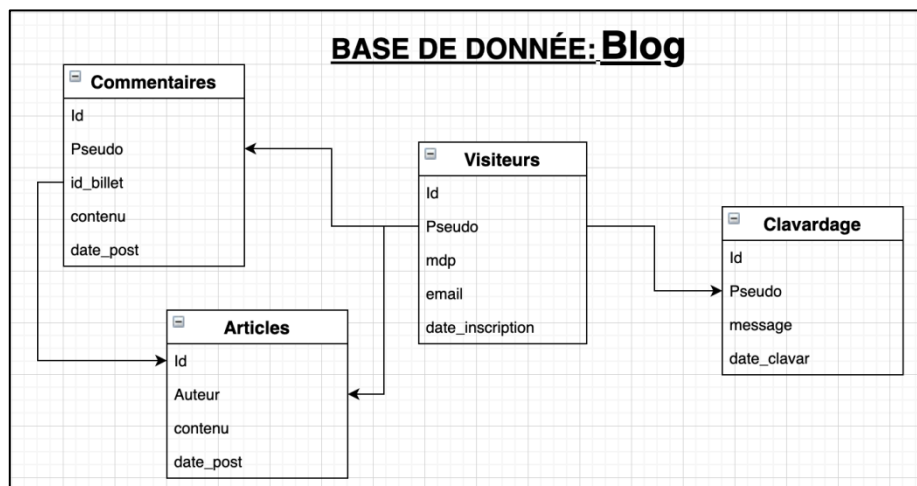
- Accueil (où l’on peut voir le logo du site)

- Blog \*
- Poster un article (page proposant de remplir un formulaire avec le titre et le contenu de l'article que l'on veut poster)
- Salon de discussion (très semblable à la page de la version minimale)
- A propos (page contenant que du html pour présenter et donner des informations sur le site)
- Ta page membre (où l'utilisateur peut voir ses informations pour l'instant que son pseudo et son adresse mail avec une option de déconnexion)

\* En ce qui concerne la rubrique blog: On peut y voir les articles postés, leur auteur, leur date ainsi qu'un lien "commentaire" sous chaque article guidant l'utilisateur vers une page avec le contenu de l'article et ses commentaires(s'il y en a). Cette page contient aussi un formulaire pour poster un commentaire sous l'article en question.

Renforcer la sécurité globale en utilisant la fonction "htmlspecialchars()" sur toutes les données fournies par l'utilisateur . Mais aussi en utilisant les conditions "if" et "else" (pour que deux comptes ne puissent pas avoir le même pseudo par exemple).

Créer de nouvelles tables SQL pour non seulement les "Visiteurs" mais aussi leur "Commentaires" et leurs "Articles" .



Pour une démonstration détaillée du code et de la structure de la base de données de cette version : regarder la vidéo jointe "Mon projet 24/03".

## C.Version finale (validation du projet)

La version finale du projet va mobiliser encore plus de travail et de connaissances. Certaines idées que j'avais prévues pour cette version ne seront pas présentes dans mon projet final par manque de temps ou parce qu'elles sont trop difficiles à mettre en place. Nous reviendront sur tout cela à la fin de cette description (qui n'inclue pas les idées abandonnées) :

Bien sur, tout ce qui à été décrit pour la version intermédiaire est applicable sur cette version

Dans la page membre:

- afficher les derniers messages (commentaires et articles) postés par nous sous forme de liens pour retrouver l'endroit où on les a postés lorsqu'on clique dessus
- afficher des informations complémentaires tels que la clé de sécurité du compte
- Un système de clé de sécurité pour réinitialiser le mot de passe de son compte (cette clé est unique pour chaque compte, composée de 6 chiffres au hasard et elle change à chaque fois qu'on change de mot de passe pour des raisons de sécurité)
- Un système de suppression pour supprimer (après confirmation) un ou des article(s) qu'on a posté

Beaucoup de petites optimisations diverses (voir vidéo du projet)

Amélioration de la page "A propos"

Des barres de recherche pour rechercher de façon intelligente des articles mais aussi des auteurs où on peut voir leur fiche publique si on clique sur leur nom

Dans la page "poster un article", on peut choisir le thème et le résumé de l'article

Dans le menu de navigation du site, l'onglet blog comporte maintenant plusieurs thèmes (sport, cuisine, autre...) pour pouvoir consulter les articles correspondants

Système de pagination pour les articles et les commentaires qui peuvent êtres parfois trop nombreux

Beaucoup optimisations de style et de sécurité

Pour le style :

Rester dans des couleurs neutres (noir, blanc, gris) mais il y a quand même des couleurs comme les liens en bleu, les titres en rouges...

Un design épuré, courbé avec beaucoup de dégradés



### Pour la sécurité:

Demande de confirmation de mot de passe à l'inscription (ou lors d'un changement de mot de passe)

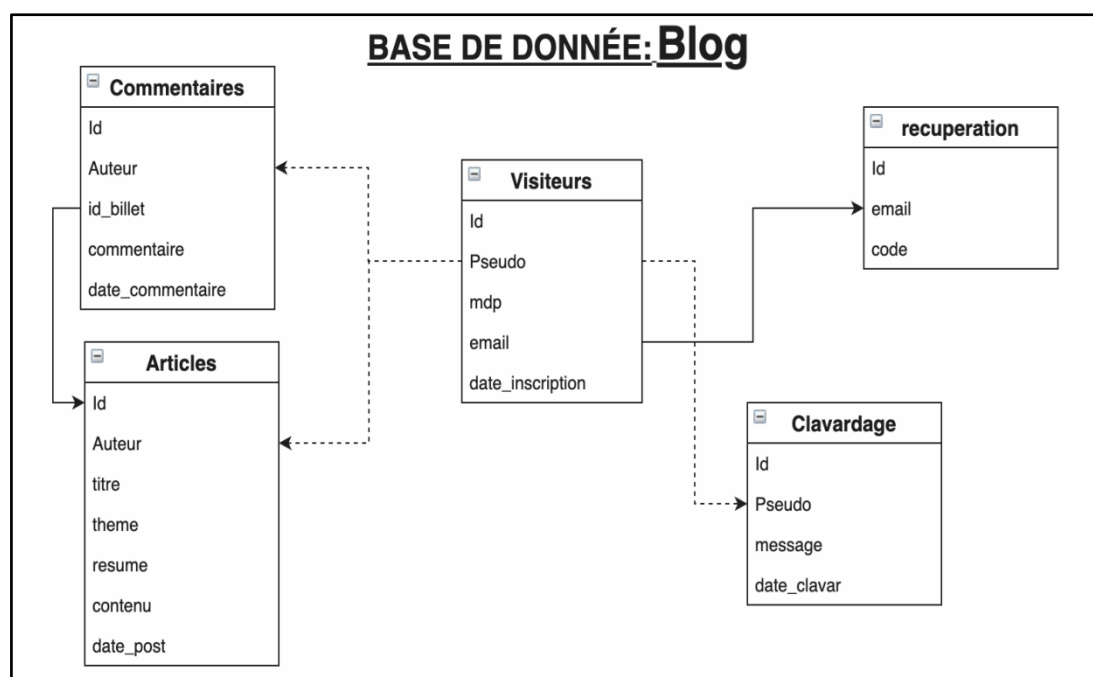
Utilisations de fonctions (php) tels que stripslashes(), addslashes(), htmlspecialchars(), password\_verify(), password\_hash(), etc... pour garantir une protection contre le maximum de failles et d'injections

Affichages d'un texte d'erreur pour l'utilisateur au cas où il modifierait mal-intentionnellement ou supprimerait des données dans l'URL

Impossibilité d'avoir une adresse email non conforme, limite de caractères (par exemple il faut un pseudo qui obligatoirement moins de 50 caractères),...

Affichages de messages pour certains formulaires comme "Pseudo trop long" ou "mots de passe non identiques"

De nouvelles tables SQL plus complexes :



**Pour une démonstration détaillé du code et de la structure de la base de donnée de cette version : regarder la vidéo jointe "Mon projet final".**

En ce qui concerne les idées qui ont été abandonnés :

- Proposer un système de "like" pour les articles et pouvoir les trier par pertinence (nombre de like en fonction de la date de publication). Cause: manque de temps

- Proposer un système de tchat privé. Cause: manque de temps et compliqué à faire. Si l'utilisateur veut parler avec quelqu'un, il peut le faire dans l'onglet "clavardage"
- Proposer à l'utilisateur d'écrire son article et ses messages avec la couleur, la police, la disposition qu'il veut (un peu comme sur Word). Cause: le travail et les connaissances nécessaires à la réalisation de cette fonctionnalité étaient trop élevés
- Proposer une option de suppression de compte ou de commentaire . Cause: Je ne voulais pas perdre mon temps sur une option que je considérais déjà comme acquise (la suppression d'article est déjà disponible). De plus, ces fonctionnalités me semblent pas très pertinentes pour la validation de mon projet

# INTERET DU PROJET

## A. Pourquoi avoir choisi ce projet ?

Nous (élèves) avons accès à toute une liste d'exemples de projets de NSI que l'on pouvait choisir. Je n'avais pas d'idée précise pour mon projet mis à part le fait que je ne voulais pas avoir recours à du matériel peu commun que je ne possède pas chez moi (robots, ect...) pour que je puisse programmer et tester mon projet n'importe où.

J'ai donc consulté la liste des projets et j'ai été étonnée du fait que 90% de ces projets sont des jeux. Je me suis donc décidée à ne pas choisir un jeu pour que mon projet soit un peu différent des autres. Étant douée dans la programmation de sites web, j'ai ainsi choisi de faire un site web avec gestion de base de donnée.

Je me suis donc renseignée sur ce à quoi pouvait ressembler un tel site et j'ai convenu que je préfèrai le faire ressembler à un blog pour qu'il soit plus attrayant et plus sympathique. Je suis donc partie sur cette base et j'ai ajouté des fonctionnalités (commentaires, récupération de mot de passe, supprimer ses articles, etc...) pour en arriver à la version finale de mon projet.

## B. Quels intérêts apporte Bloggy ?

Bloggy permet à plusieurs utilisateurs de communiquer entre eux et de s'échanger des informations. Ils peuvent pour ce faire, poster des articles, les commenter ou discuter en ligne dans un onglet "salon de discussion". Chaque personne bénéficie de son espace membre où il peut visualiser certaines informations relatives à son compte mais aussi supprimer les articles qu'il a posté. Le site dispose aussi d'un outil de recherche d'utilisateur (grâce au pseudo) qui une fois trouvé, affiche les informations publiques de ce dernier.

L'intérêt principal d'un tel projet, est donc de pouvoir fournir à des utilisateurs une interface dynamique où ils peuvent s'exprimer librement (via le site web). Chacun peut découvrir les posts des autres, les commenter, discuter ensemble. Nous pouvons rapprocher Bloggy à une sorte de réseau social (très simplifié) qui rapproche les personnes et leur permet de s'épanouir. Le public visé est donc très large vu que le thème du site dépend de ce que ses membres peuvent publier.

La seule contrainte est que l'on est obligé de créer ou de posséder un compte valide pour pouvoir accéder au contenu du site.

# SUIVI DU PROJET

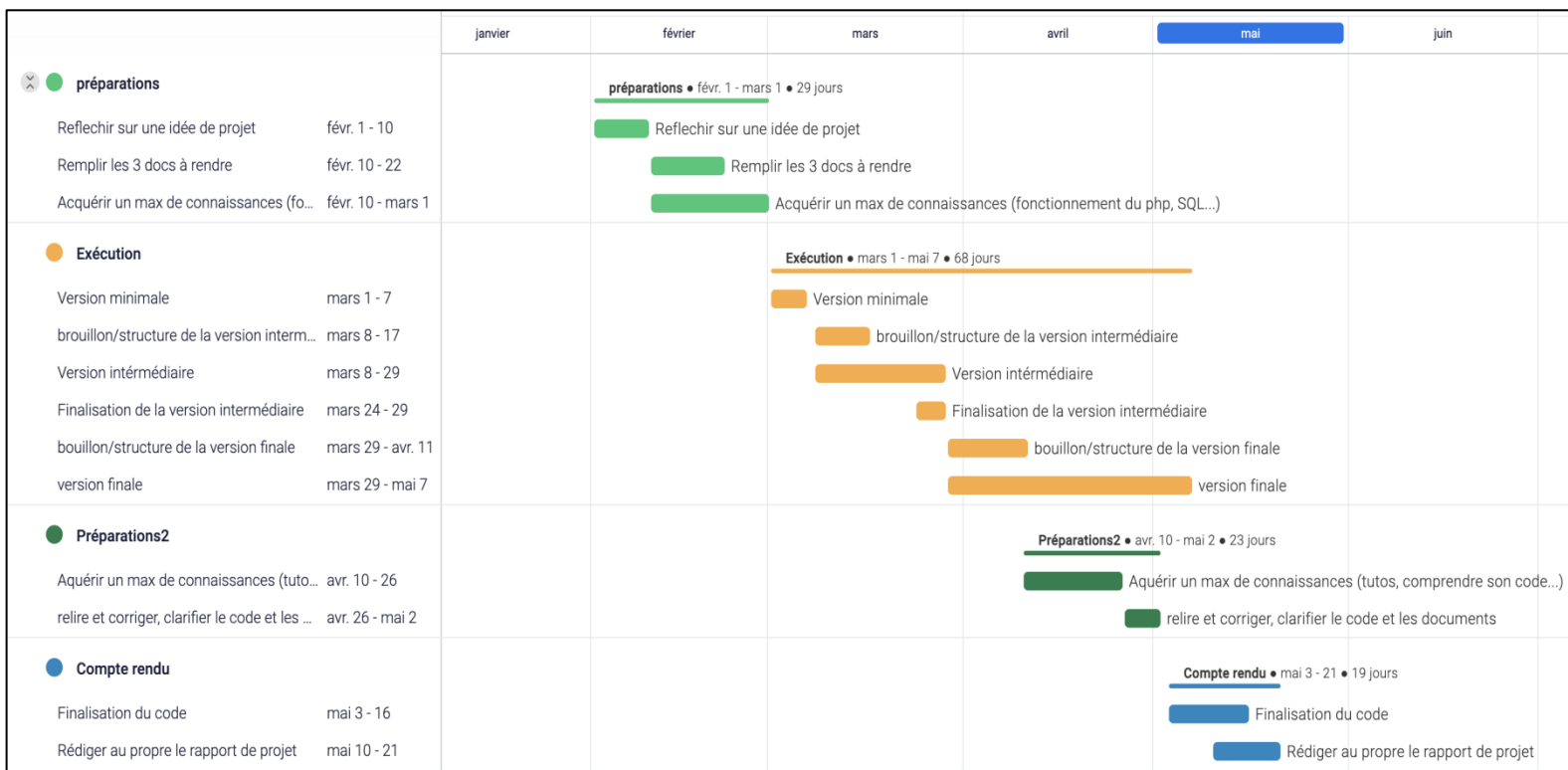
## A. Planning prévisionnel

Pour réussir un projet, il est essentiel de bien mettre en évidence l'ensemble des tâches à réaliser en un temps donné.

Grâce à un planning prévisionnel, il a été possible de fixer des dates pour lesquelles certaines parties de ce grand projet doivent être terminées.

Diagramme de Gantt:

C'est un outil efficace pour avoir une vue globale des tâches à réaliser et de l'avancement du projet au cours du temps



Mon planning initial était d'acquérir les connaissances nécessaires pour mon projet pendant les vacances et de le réaliser pendant les semaines où l'ont avait cours de NSI. Ainsi il était prévu que je finisse le projet initial en deux semaines (ça m'a pris moins d'une semaine), que je finisse la version intermédiaire avant avril (je l'ai fini le 29 mars) et que je continue sur la version finale jusqu'à la fin de l'année. Sans compter les documents que je complétais au fur et à mesure que mon projet avançait.

Vu que je ne savais pas la date exacte à laquelle il fallait rendre le projet, je

me suis imposé des relectures fréquentes de mon code pour être prête à le rendre à tout moment.

J'avais prévu qu'il m'était impossible de finir la version finale de mon projet dans le temps imparti. J'ai donc dû me fixer des priorités (quelle partie est plus importante que l'autre) pour que le rendu final soit cohérent.

Nous pouvons remarquer que ce planning n'est pas précis et c'est tout à fait normal. Mon rythme de travail était régulier mais il n'en est pas de même avec les tâches réalisées. Mes seules dates importantes sont la finalisation des versions minimales, intermédiaires et finales. Je me permettais donc d'être libre le reste du temps à condition que je ne sois pas en retard. Il était très rare que je me consacre sur une chose à la fois lors de la réalisation de mon projet. Je créais/modifiais le site, les bases de données, le style, les documents, etc.. en même temps. Le fait de travailler sur de nombreuses choses à la fois et en même temps peut s'expliquer par le fait que je sois seule dans mon équipe mais ce n'est pas une mauvaise chose.

## **B. Gestion des aléas**

Comme tout projet, j'ai dû faire face à des aléas. Cette partie peut être jointe à la partie "B. Des contraintes et leurs solutions" dans "programmations du projet" où j'y expose toutes les erreurs et soucis techniques que j'ai dû affronter.

Concernant les contraintes extérieures au projet :

Épreuves et examens:

- Bac blanc le 9 novembre (écrit) et le 6 février (oral)
- Bac blanc n°2 le 10 mai (écrit) et le 28 mai (oral)
- Olympiades de mathématiques le 23 mars

## **C. Temps consacré à la réalisation**

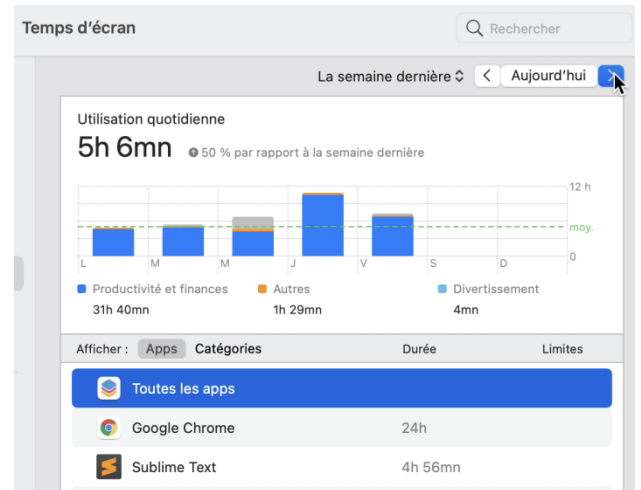
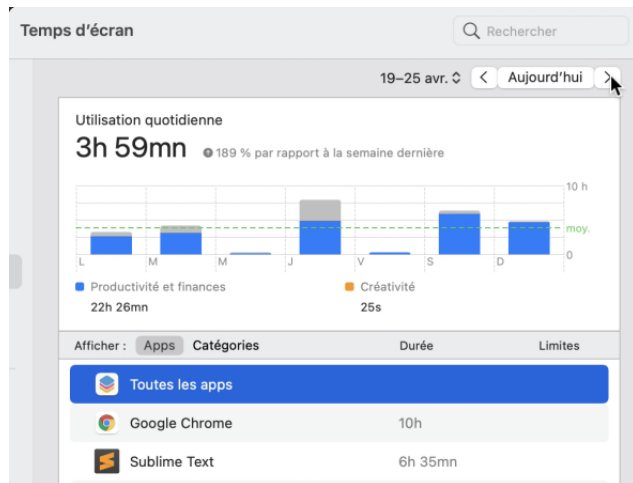
Comme il a été spécifié dans le diagramme de Gantt, la durée de ce projet s'étale de début février à fin mai. Nous allons voir ce que cela donne en terme d'heures:

- Nombre d'heures de travail pendant les cours: 2 à 4 heures par semaine soit au total environs 28 heures (il y a eu beaucoup de jours fériés et vacances)
- Nombre d'heures de travail en dehors des cours: +80h pendant les vacances + environs 2/3h par semaine soit au total 100 heures

- Total: Il m'a fallu 128 heures de travail pour réaliser mon projet

(Il m'a fallut plus de 12h rien que pour rédiger ce document)

PS: Je sais le temps exact que m'a pris mon projet grâce à la fonctionnalité "temps d'écran".



# PROGRAMMATION DU PROJET

## A. Organisation des pages web

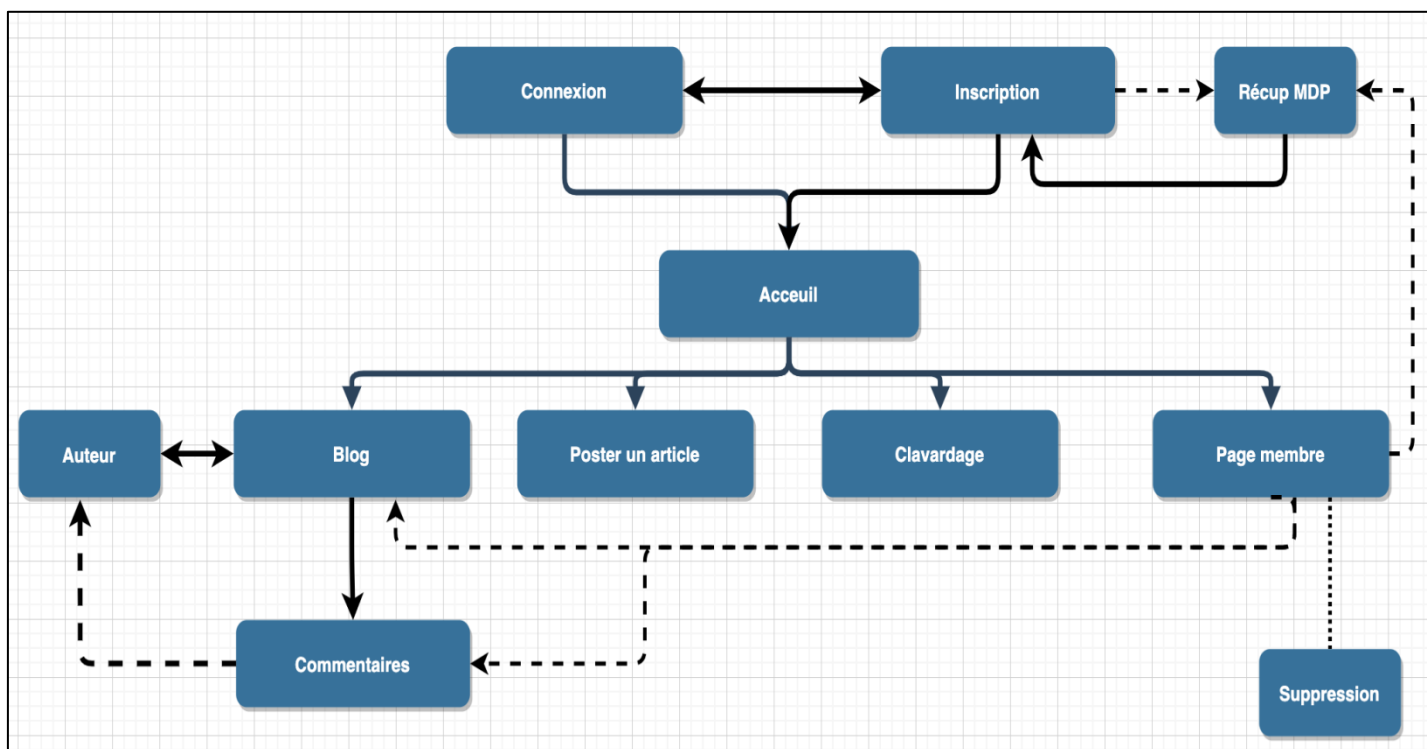
Lorsque l'on développe un site web qui comporte plusieurs pages ou fichiers nécessaires à son organisation, il est important de bien savoir la structure globale du site avant de se lancer dans la programmation.

La version minimale de mon projet ne comporte qu'une seule page html pour l'affichage accompagnée d'une page php pour le traitement. Dans ce cas, l'organisation est simple et pas importante.

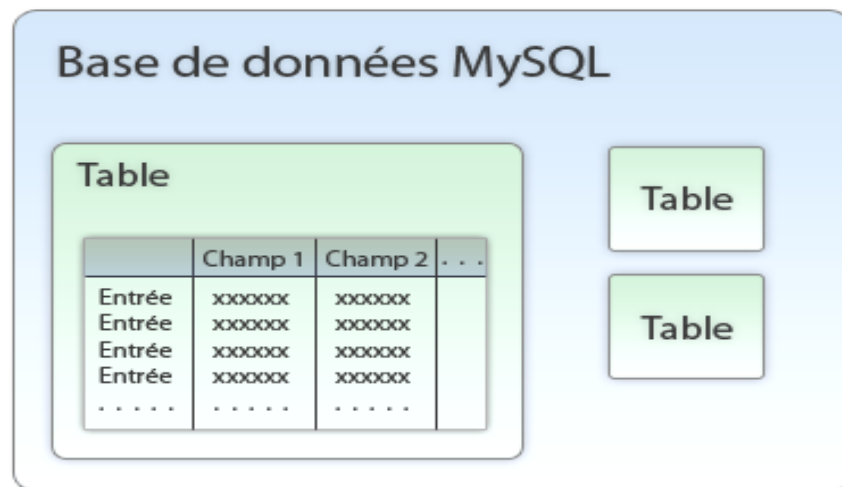
En revanche, la version finale de mon projet fait appel à de nombreuses pages et fichiers divers du fait de sa complexité. Se retrouvant facilement avec plus d'une dizaines de fichiers pour un même site. Il faut veiller aussi à soigner l'arborescence de ce dernier (voir image ci-dessous). Cela permet d'améliorer l'efficacité du site et l'expérience de utilisateur.

Pour limiter le nombre de fichiers et les liens entre eux, j'ai essayé d'incorporer le code PHP,SQL, HTML de chaque page dans un seul fichier. Ainsi, (presque) chaque fichier peut se considérer comme autonome et n'a pas besoin d'un autre fichier de traitement à part pour afficher les bonnes informations à l'utilisateur.

Schéma d'arborescence de mon site final\* :



\* Toutes ces pages sont liées à un fichier de style "style.css"



## B. Structure de la base de donnée

L'une des parties les plus emblématique des sites web avec gestion de base de données est l'organisation et la structure de cette dernière et de ses tables qui peuvent devenir un vrai casse tête. Il est vraiment indispensable d'avoir fait un brouillon avant de se lancer dans ce genre de projet sinon la base de donnée risque d'être mal organisée et/ou compliquée à comprendre et à interpréter.

Lorsqu'on n'a besoin que d'une seule table comme avec la version minimale du projet, cela ne devrait poser aucun problème. On retrouve souvent les mêmes structures d'un site à un autre. Cependant, la version finale est bien plus complexe. Il est nécessaire de lier certains champs de certaines tables avec d'autres, de créer des champs spécifiques (par exemple `id_billet` dans Commentaires) pour faire les jointures entre les différentes tables... Tout cela en essayant d'optimiser au maximum la base de donnée.

Pour savoir manier tout cela, il faut faire appel à un nouveau langage: le SQL. Ce langage permet d'effectuer des actions sur une base de donnée. Par exemple, rechercher, ajouter, modifier ou supprimer certaines données qui peuvent être triés selon un ordre précis.

Dans mon projet, ce langage est complémentaire au PHP. En effet, le PHP m'a permis d'analyser et de vérifier les données fournies par l'utilisateur grâce à des conditions ou des fonctions tandis que le SQL m'a permis d'interagir entre les données traités par PHP et la base de donnée ou à l'inverse, de fournir les données nécessaires qui sont traitées par PHP (et affichés sur le site)



La structure schématisée de la base de donnée que j'ai utilisé pour mon projet se situe dans la partie "description du projet" de ce document.

## C. Logiciels et langages utilisés

J'ai du faire appel à beaucoup de langages et de logiciels nécessaires à la création de mon projet.

Langages utilisés: Du html, du CSS, du PHP, du SQL et un peu de JavaScript.

Logiciels nécessaires:

- Un logiciel de traitement de texte (sublime text 3)
- WAMP, MAMP ou LAMP contient tous les outils nécessaires pour pouvoir coder du php (un serveur Apache, MySQL et PHP)
- Un navigateur web (j'en ai utilisé plusieurs) pour pouvoir tester le code

En ce qui concerne le matériel technique, j'ai bien évidemment besoin d'un ordinateur avec ces logiciels installés.

## D. Les contraintes et leurs solutions

Comment coder un programme sans erreur ? Impossible.

Au cours de la programmation de mon projet, j'ai bien sur dû faire face à de nombreuses erreurs et problèmes techniques. Mettant parfois ma patience à rude épreuve. Mais ces erreurs m'ont permis de mieux comprendre et de progresser dans mon code.

Tout d'abord, l'installation de WAMP. Après avoir fini sont installation, l'application m'affichait un message d'erreur très très long :



Après avoir rechercher la cause de l'erreur, je me suis rendu compte qu'il fallait que j'installe TOUS les paquetages qui étaient listés sur ce message d'erreur car ils étaient essentiels au bon fonctionnement de l'application (mon ordinateur ne les avait pas car il est un peu trop vieux). Après 11h de recherches et téléchargements, le même message d'erreur s'affiche sauf qu'il est (un peu) plus court. J'essaye donc de revérifier mes fichiers, de réessayer... Bref, rien ne semble marcher et ça faisais des jours que je cherchais une solution.

Solution: Je décide de changer non pas d'ordinateur mais d'ordinateur ET de système d'exploitation. Et MAMP à mis beaucoup moins de

temps pour marcher correctement (même s'il y a eu quelques petits soucis)

Les messages d'erreur de php ne s'affichaient pas non plus au début. Ce qui fait que je ne savais pas pourquoi mon code ne marchait pas et ce n'était pas normal. Solution: changer de moteur de recherche ou ajouter une commande dans le fichier info.php (ce que j'ai fais).

Lorsque j'ai voulu configurer la fonctionnalité de récupération de mot de passe par email (après avoir suivi un cours pour savoir comment faire), impossible de pouvoir envoyer d'email avec mon site. Solution: j'ai tout essayé pendant des jours sans succès, j'ai donc du abandonner cette idée. Pas de solution




La majorité des erreurs que je faisais au niveau de la programmation de ce projet étaient les erreurs de syntaxe (manque une parenthèse, un point-virgule..)

Solution: bien relire son code

# ANNEXES

## A. Sources utilisés pour l'apprentissage

- Les cours de NSI vu tout au long de l'année et les publications sur classroom
- OpenClassroom:

Cours	Inscription	Progression
Apprenez à créer votre site web avec HTML5 et CSS3	02/12/2020	93 % 
Apprenez à programmer avec JavaScript	09/04/2021	95 % 
Concevez votre site web avec PHP et MySQL	23/02/2021	97 % 

- De nombreuses vidéos tutoriels surtout de PrimFX.com et Grafikart.fr
- Des sites de documentation: php.net ou w3schools.com

## B. Autres sources

Pour réaliser les diagrammes et les schémas:

- Draw.io
- Monday.com

Pour l'inspiration :

- Sites web divers, surtout des blogs comme <https://www.plussimple.fr/actualites/>
- Mes projets et exercices précédents notamment le site web que l'on devait créer au cours de l'année
- les exercices/TP des cours qui étaient proposées sur Openclassroom
- Les documents, liens, cours publiés sur le classroom de NSI tout au long de l'année

# CONCLUSION

Ce projet à été une expérience enrichissante pour moi.

En effet, cela m'a permis de mettre en pratiques les connaissances que j'avais vues en cours et en dehors mais aussi d'apprendre beaucoup de nouvelles notions. J'ai dû m'organiser sur une grande période et veiller à ne jamais être en retard lors de la réalisation de ce projet même si le temps nécessaire à la conception de certaines parties était difficile à estimer.

Se lancer seule peut faire peur au début mais en me fixant des objectifs, j'ai réussi à prendre conscience de l'importance de chaque parties de ce projet. J'ai aussi été confrontée au fait que pour programmer, il ne suffit pas de coder mais il faut aussi structurer, planifier et organiser son travail.

Le fait que le projet soit à rendre dans un temps imparti m'a obliger à penser et adapter la forme de mon projet en fonction du temps que j'allais avoir pour le réaliser. Mais si je disposais de plus de temps, Bloggy aurait pu devenir en plus d'un blog, un vrai mini réseau social avec beaucoup plus de fonctionnalités.

Cette expérience m'a aussi permis de voir le monde informatique sous un nouvel angle. J'ai eu l'occasion de me mettre à la place d'un développeur et je ne me serais jamais doutée que pour créer un "simple" site web, il fallait faire appel à autant de connaissances et de patience.

J'ai rarement eu à réaliser quelque chose de si complexe et long au cours de ma scolarité. Le fait d'être amené à développer un tel site de A à Z est représentatif du travail que l'on pourrait réaliser dans la vie professionnelle. De plus, la réalisation finale est vraiment quelque chose dont je suis fière.

Une bonne expérience à renouveler !