



UNIVERSITÉ DE  
MONTPELLIER



FACULTÉ DES SCIENCES  
DE MONTPELLIER



Réseau **Figure**  
COMPRENDRE • EXPLORER • TRANSFORMER

## UM.tar Rapport Projet CMI

Marc-Antoine MOUTTET, Arthur DE FAYS, Aurélien BIDAUT

Janvier-Avril 2023

# Table des matières

1	Introduction	2
1.1	Projet CMI . . . . .	2
1.2	Notre projet . . . . .	2
2	Cahier des charges	3
2.1	Objectifs . . . . .	3
2.1.1	Mk1 . . . . .	3
2.1.2	Mk2 . . . . .	3
2.1.3	Mk3 . . . . .	3
2.1.4	Mk4 . . . . .	3
2.1.5	Mk+ . . . . .	3
2.2	Sécurité . . . . .	3
3	Planification du Projet	5
4	Sources/Liens	6

# 1 Introduction

## 1.1 Projet CMI

Dans le cadre du second semestre de notre cursus universitaire (du diplôme CMI Informatique), nous devons élaborer un projet en groupe.

Le but est de savoir s'organiser en groupe pour définir les objectifs et les tâches de notre projet. Il doit aussi être de notre niveau.

De plus, il doit nous permettre de nous rendre compte que nous sommes capable de faire un projet en groupe en autonomie.

## 1.2 Notre projet

Notre projet se base sur un besoin de centraliser les cours, fiches, exercices et corrections des Unités d'Enseignement en informatique.

En effet, nous nous sommes rendu compte lors des révisions des partiels du premier semestre de cette année, qu'il nous manquait des ressources pour s'entraîner. Nous avons aussi constaté qu'il serait intéressant d'avoir un endroit pour centraliser tous les cours, donc faciliter les révisions.

Nous avons donc eu l'idée, de créer ce site, qui concentrera les fiches de cours, mais aussi des contrôles ou partiels pour pouvoir réviser aisément.

Langages utilisés :

- Markdown pour l'écriture des fiches de révision et des exercices.
- PHP pour le côté serveur du site, qui permettra de faire des requêtes SQL et contrôler l'accès au site.
- SQL pour les requêtes sur la base de données des utilisateurs et des exercices.
- LaTeX pour rédiger le rapport.
- HTML, Javascript et CSS pour la mise en page, l'envoi des données et le style du site.

## 2 Cahier des charges

### 2.1 Objectifs

#### 2.1.1 Mk1

- Utilisation d'un alias de l'adresse mail principale du site internet.  
L'email `verify@mma.dev` est utilisé pour envoyer les mails de vérification.
- Envoie un email avec code de vérification pour valider l'adresse mail universitaire de l'étudiant.
- Le compte utilisateur est ensuite vérifié et peut accéder à la totalité du site.
- L'authenticité de l'utilisateur est vérifié à chaque requête.
- Amélioration du système de navigation dans le site.
- Mise en place de la mise en page et du CSS.

#### 2.1.2 Mk2

- Ajout d'exercices de base et de leurs corrections.
- Ajout de fiches de cours.
- Ajout d'un système d'upload des exercices via une interface afin de faciliter la mise en BDD.

#### 2.1.3 Mk3

- Possibilité d'enregistrer les fiches générées dans un onglet à part.

#### 2.1.4 Mk4

- Création d'un programme qui créer des fiches de partiel ou de contrôles, avec des valeurs aléatoires, et de leurs corrections respectives.

#### 2.1.5 Mk+

- Pour chaque exercice, mettre en place un barème et une note.

### 2.2 Sécurité

Le site a tout d'abord un aspect de sécurisation des données de ses utilisateurs et du contenu mis à disposition très important. Pour cela, le site utilise un certificat TLS/1.3 de la société Let's Encrypt, afin de garantir la sécurité des données échangées entre le serveur et le client tel que les mots de passe, les adresses email, etc.

Le serveur et le client utilisent un protocole de communication sécurisé, le HTTPS, qui permet de chiffrer les données échangées. Cette sécurité est nécessaire afin de protéger le serveur et le client d'être victimes d'attaques. Par exemple : Man In The Middle (MITM), en français "L'Homme du milieu", qui consiste à intercepter les données échangées entre le serveur et le client, afin de les modifier ou de les voler.

Nous avons décidé de limiter la création de compte aux étudiants rattachés à l'Université de Montpellier, ayant donc une adresse mail du type `etu.umontpellier.fr`.

Afin de protéger toutes les pages du site, celles-ci, dès lors de leur requête par un utilisateur au serveur, exécutent un fichier PHP permettant de vérifier l'existence et l'authenticité de la session.

Pour cela, nous utilisons `session_start()` qui nous permet de démarrer une session. Nous vérifions si les variables de session dont nous avons besoins existent, comme :

- Le timestamp de création de session est utilisé pour vérifier si la session n'a pas expiré. *Une session expire toutes les 24 heures, ceci afin d'assurer la sécurité du site et des utilisateurs.* Les autres variables de sessions servent afin que l'utilisateur puisse utiliser le site sans avoir à se reconnecter à chaque fois. Mais ces variables servent aussi à vérifier l'authenticité de l'utilisateur.
- L'email permet dans un premier temps de vérifier si le compte existe bel et bien.
- Le hash du mot de passe permet de vérifier si le mot de passe est valide, c'est à dire que celui de la session va être comparé à celui de la base de données, s'ils sont identiques alors c'est bon. Le hash de mot de passe a un rôle similaire à un token : en cas de changement de mot de passe, celui-ci change et rend invalide toutes les autres sessions.
- L'ID d'utilisateur
- L'ID de session est aussi envoyé au serveur, afin de vérifier si l'ID de session est valide. Ceci permet de limiter le nombre de session simultanées d'un utilisateur à une seule afin d'améliorer la sécurité du site. Si une de ces variables n'existe pas ou n'est pas valide, l'utilisateur est redirigé vers la page de connexion. Si toutes les variables sont valides, l'utilisateur peut accéder à la page demandée. L'ID de session est régénéré à chaque fois que l'utilisateur se connecte et il est mis à jour dans la base de données, afin d'éviter les attaques de type fixation de session. *L'attaque par fixation de session consiste à forcer l'utilisateur à utiliser et entrer ses données sur une session que l'attaquant a créé, l'attaquant peut ensuite utiliser le numéro de session pour usurper l'utilisateur.*

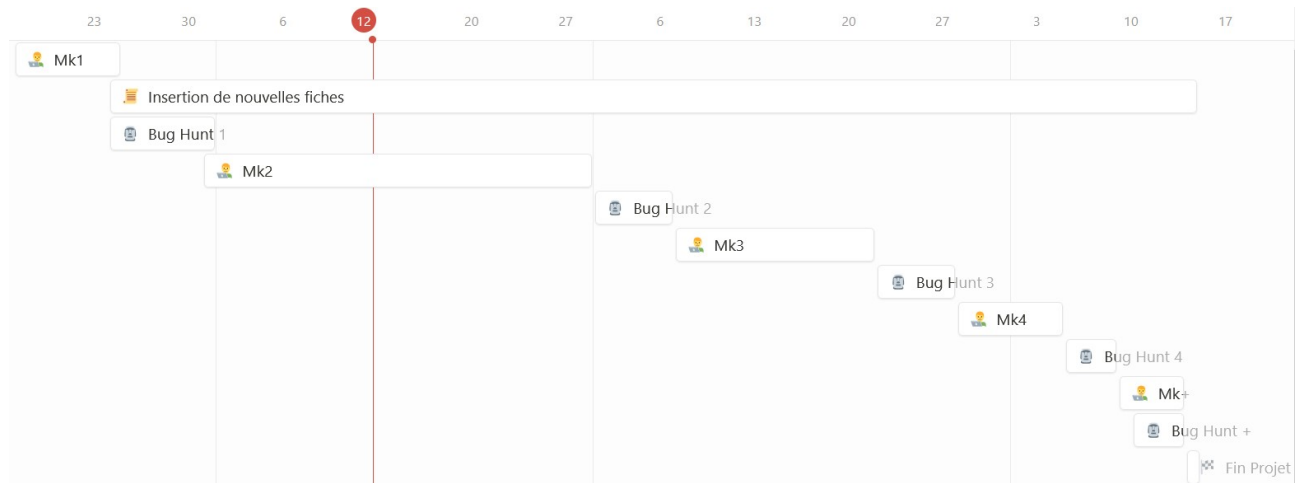
### 3 Planification du Projet

Mk → étapes du projet.

Bug hunt → étapes de recherche des bugs ainsi que de les résoudre.

- 17-24 janvier : Mk1
- 24-31 janvier : Bug hunt 1
- 1-28 février : Mk2
- 1-6 mars : Bug hunt 2
- 7-21 mars : Mk3
- 22-27 mars : Bug hunt 3
- 28 mars - 4 avril : Mk4
- 5-8 avril : Bug hunt 4
- 9-13 avril : Mk+ si on a le temps.

**La planification peut changer au fur et à mesure : si l'on finit un objectif en avance ou si nous considérons qu'il nous faut plus de temps afin de valider une étape. À partir du Mk2, nous avons décidé d'ajouter du temps afin d'apprendre le PHP, JavaScript et autres langages utilisés dans le projet.**



## 4 Sources/Liens

Lien de la page principale du site : <https://mma.dev/um/>.

Lien GitHub : <https://github.com/TheGeeKing/UM.tar-Rapport-CMI-L1>.

ChatGPT, OpenAI, <https://chat.openai.com/chat>.