#### Site to do list réalisé par :

- -Meradji Anaïs
- -Gacem Aicha

#### Task Master



Organisez vos tâches, pour une vie plus paisible



#### INTRODUCTION AU SITE

OBJECTIFS PRINCIPAUX DU SITE WEB

GITHUB/ ASPECT COLLABORATION

FRAMEWORKS UTILISÉS ET OUTILS

PROTOTYPE SITE WEB

# Introduction au site web

#### introduction

Pourquoi le nom de site Task Master?

Task Master





Tâche

Maitre

Maitre des tâches



#### introduction

Task Master est un site web qui permet de réaliser des to do list ainsi que de les classifier par types, ainsi que par date. Il permet également de gérer les tâches prioritaires en les mettant plus en avant parmi les tâches grâce à un code couleur.



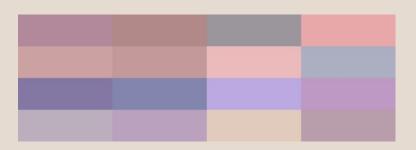


#### Objectifs principaux du site web

• Présenter une plateforme conviviale et attrayante.

Pour cela nous avons eu recours à **Figma**, un éditeur graphique, afin de visualiser l'aspect du site.

La palette de couleurs a été sélectionnée selon nos préférences mais toujours dans le but d'aboutir à un site plaisant visuellement, que le cerveau n'interprètera pas comme stressant.



#### Objectifs principaux du site web

- Faciliter la gestion des différentes tâches.
- Permettre de mener une vie plus organisée et donc moins stressante et plus productive.
- Avoir une idée claire de son planning de la semaine.
- Eviter d'oublier les tâches importantes.
- Aider l'utilisateur à améliorer sa vie personnelle et professionnelle.



#### C'est quoi le GIT?

Git est un système de gestion de versions décentralisé utilisé pour le suivi des modifications apportées au code source d'un projet informatique.

#### C'est quoi GITHUB?

GitHub est une plateforme en ligne de gestion de versions et de collaboration pour le développement de logiciels.



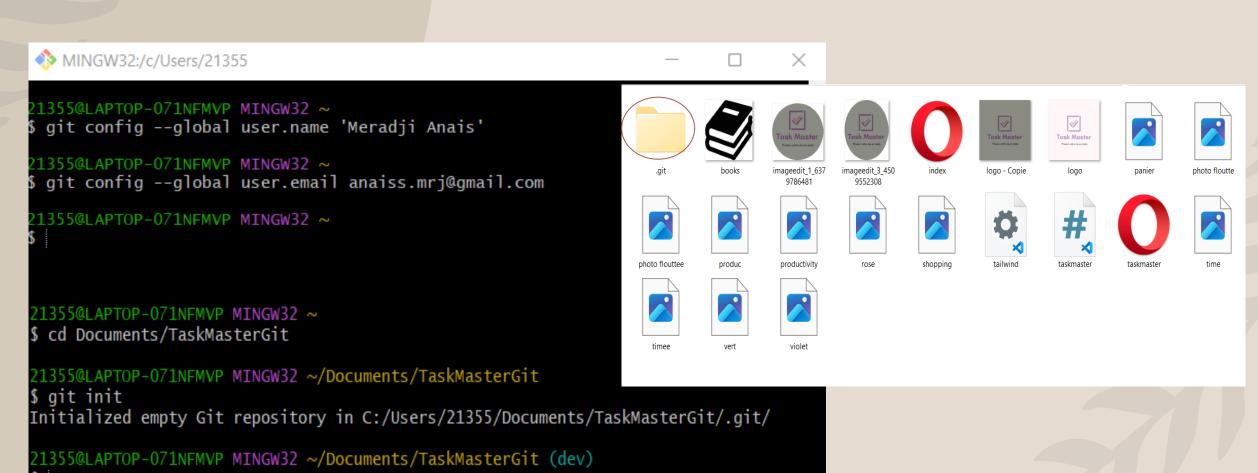
#### Comment nous avons utilisé GitHub

Etape 1 : Déposer notre projet dans une branche à travers l'outil VS code

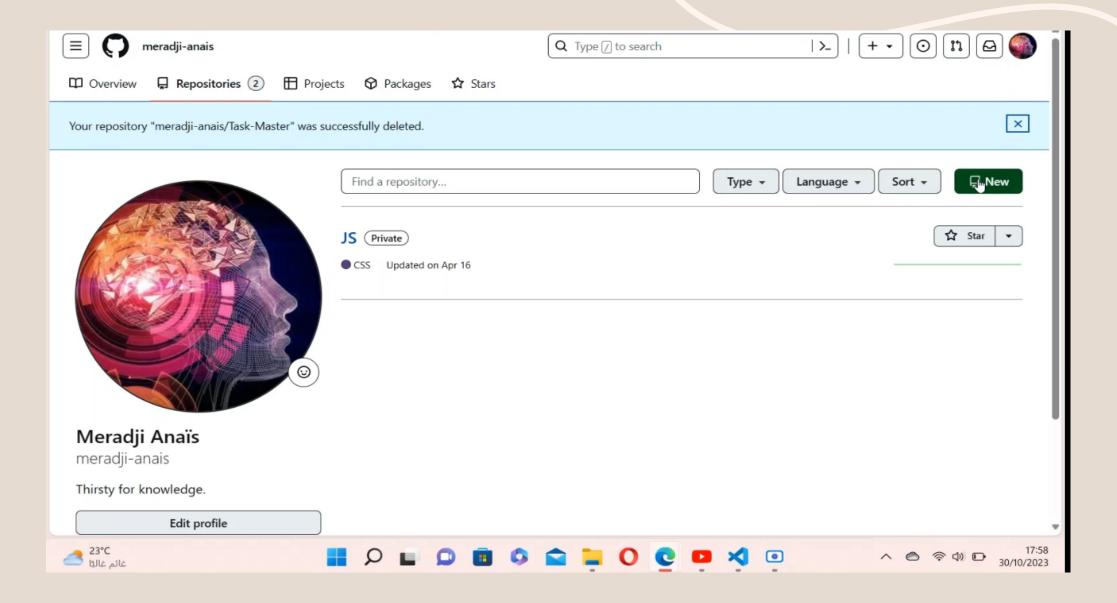
Etape 2 : Donner l'accès au binôme

Etape 3 : Commandes de modification

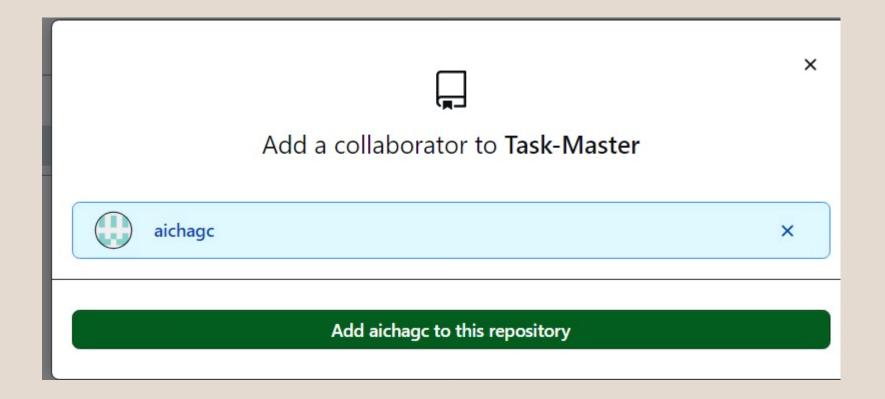
#### Créer un repository local



#### Déposer son projet sur GitHub



#### Donner l'accès au binôme



#### Commandes de modification

A travers le terminal de vs code, voici les commandes à taper afin de push un élément dans le repository GitHub, et donc avoir la nouvelle version.

```
PS C:\Task Master> git add taskmaster.css
PS C:\Task Master> git commit -m 'new design'
[dev 1a260b8] new design
   1 file changed, 27 insertions(+), 11 deletions(-)
PS C:\Task Master> git push origin dev
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 685 bytes | 97.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/meradji-anais/Task-Master.git
   38d65b2..1a260b8 dev -> dev
PS C:\Task Master> []
```

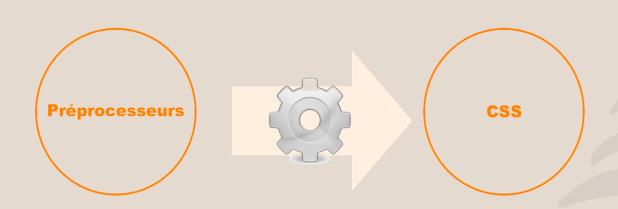
#### Commandes de modification

	7 <del>7</del>		
shop	ping.jpg	init	last week
tailwi	ind.config	init	last week
taskn	master.css	new design	1 minute ago
🖰 taskn	master.html	mend	last week
time.	png	init	last week
timee	e.png	init	last week

# Frameworks utilisés et outils

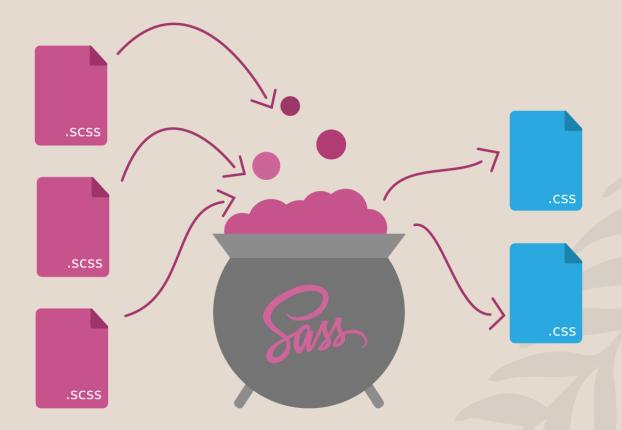
#### Un aperçu rapide du framework SASS et de ses avantages:

SASS est un préprocesseur CSS qui simplifie et améliore l'écriture des feuilles de style. Il offre des fonctionnalités avancées, comme les variables, les boucles, et les mixins, facilitant ainsi le développement front-end.



#### Pourquoi utiliser SASS?

SASS rend le CSS plus efficace et maintenable. Il permet d'économiser du temps en éliminant la répétition de code et en facilitant les modifications. Ses fonctionnalités avancées améliorent la lisibilité et l'évolutivité des feuilles de style.



### Principales fonctionnalités de SASS

Les fonctionnalités clés de SASS incluent les variables, les sélecteurs imbriqués, les mixins, les fonctions, et les importations. Ces caractéristiques améliorent la modularité et la réutilisabilité du code CSS.

#### Exemple de Code SASS

Les variables nous permettent de prédéfinir des styles particuliers que nous utiliserons dans notre site. Pour créer une variable, il suffit de mettre un \$ suivi du nom de la variable :

```
    variables.scss X

sass > base > ? variables.scss > ...

    //Déclaration de la variable

2
3    $primary : □rgba(163, 247, 9, 0.2);
```

Ici le concept est de pouvoir imbriquer, enchainer les règles css les unes dans les autres pour pouvoir éviter la répétition

```
# style.css ×
                                                                       sass > 8 style.scss > 4 .maListe > 4 ul > 4 li > 4 a
      .maListe {
        background-color: ■red;
                                                                              .maListe{
                                                                                  background-color: ■red;
      .maListe ul {
        height: 50px;
                                                                                  ul{
        width: 100px;
                                                                                      height: 50px;
                                                                                      width: 100px;
      .maListe ul li {
                                                                                      li{
        list-style-type: none;
                                                                                           list-style-type: none;
        padding: 5px;
                                                                                          padding: 5px;
      .maListe ul li a {
        text-decoration: none;
                                                                                               text-decoration: none;
```

Le principe des mixins va tout simplement nous éviter de répéter un bloc de code elles peuvent s'avérer très utile suivant les situations dans lesquelles nous nous trouvons. La meilleure utilisation possible est de créer un fichier dans notre dossier base qui contiendra notre mixin. Une fois la mixin crée nous allons faire un import de fichier dans notre fichier principal et la déclarer comme ceci <@include maMixine>

```
mixins.scss X
sass > base > 9 mixins.scss > 🕥 cardStyle
       @mixin cardStyle{
           box-shadow: 2px 2px 2px ■#000;
           border-radius: 5px;
           padding: 15px;
           margin: 15px;
           background-color: ■#ccc;
           color: ■#000;
           height: 20%;
           width: 20%;
```

Task Maker 24

## Comment intégrer SASS dans votre workflow?

Pour intégrer SASS dans votre workflow, vous pouvez utiliser des outils de compilation tels que Node.js avec npm, Ruby Sass, ou des plates-formes de développement web comme CodePen. Ces outils transforment votre code SASS en CSS standard pour une utilisation web.

#### LES OUTILS UTLISÉS:



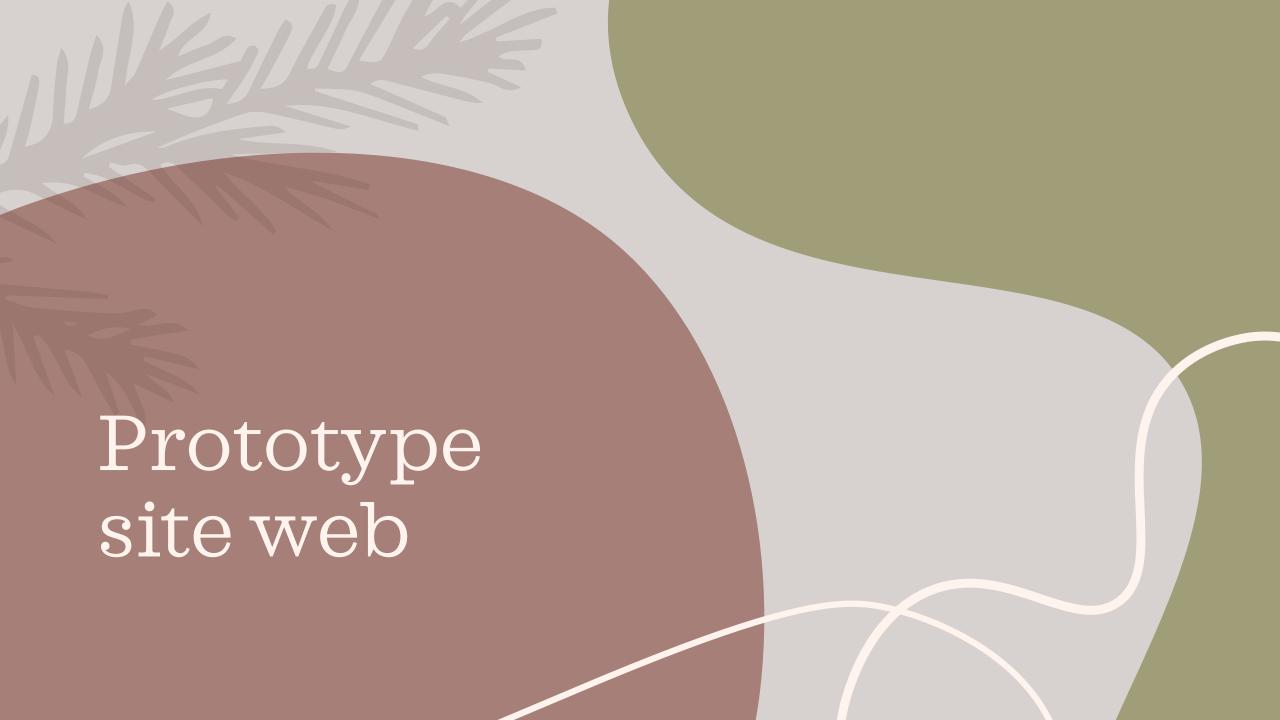
Visual Studio Code (VS Code) est un éditeur de code source gratuit et open-source.



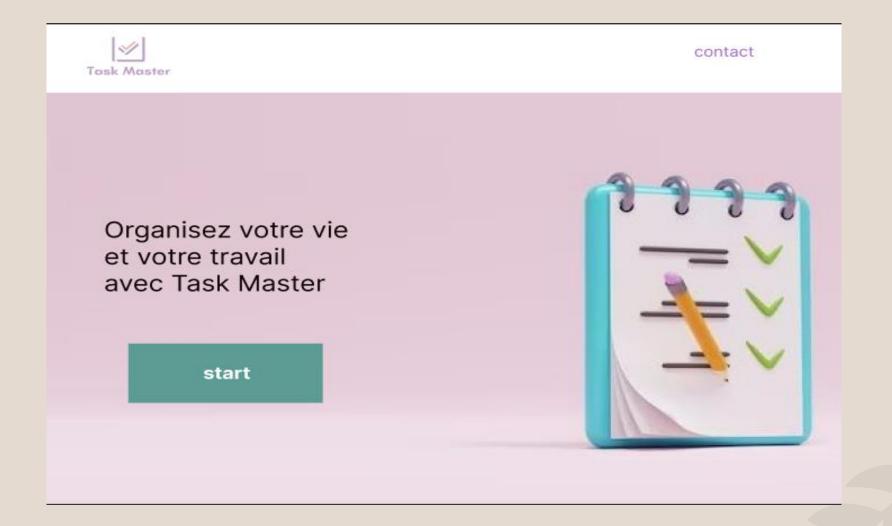
Figma est un outil de conception d'interface utilisateur basé sur le cloud qui permet aux équipes de créer, de collaborer et de prototyper des designs interactifs.



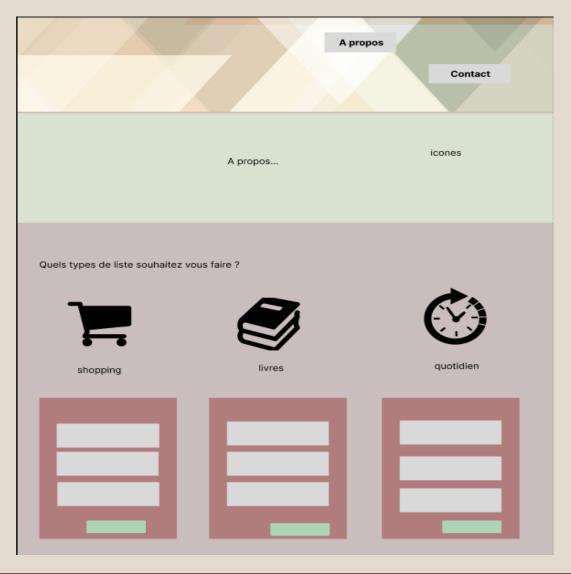
GitHub est une plateforme de développement collaborative basée sur Git, un système de gestion de version

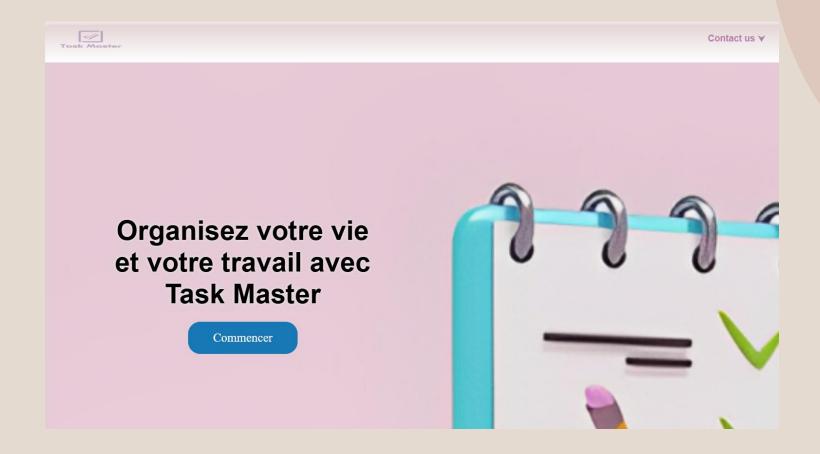


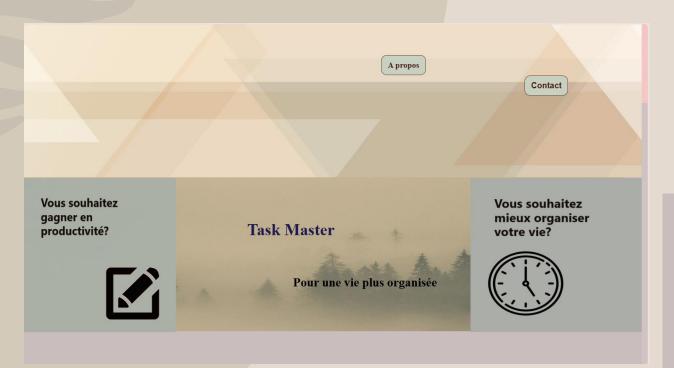
#### Prototype site web



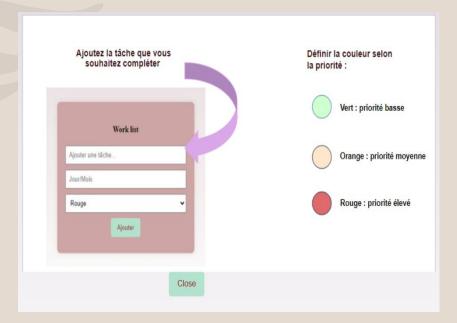
#### Prototype site web













Shopping lis	t		<u>Work list</u>
Ajouter une tâche		A	jouter une tâche
Jour/Mois		Jo	our/Mois
Rouge	~	V	/ert ~
Ajouter			Ajouter
acheter une jupe	08/04		envoyer un mail 03/04
acheter du lait	04/04		

