

Travail collaboratif sur la maquette (Projet en cours)

Objectif :

Vous travaillez en groupe sur une **maquette fonctionnelle d'un projet web**.

Chaque membre de l'équipe est **responsable d'une partie distincte : au minimum une page complète**, réalisée selon les normes vues en cours.

Travail demandé

- Chaque étudiant doit livrer **une page maquettée**, avec au minimum :
 - Un ou deux **effets visuels simples** (ex. : hover , transition, animation).
- La mise en page doit rester cohérente avec la **charte du groupe** (couleurs, typographie, navigation).

Livrables attendus

1. Un **prototype** (wireframe ou capture d'écran annotée).
 2. Un dépôt Git contenant :
 - Un README.md avec :
 - le **lien de la maquette en ligne** (Figma ou image),
 - une **capture d'écran** de la page,
 - une courte description du rôle de la page.
-

Challenges

Challenge 1 — Identifier les balises d'un wireframe

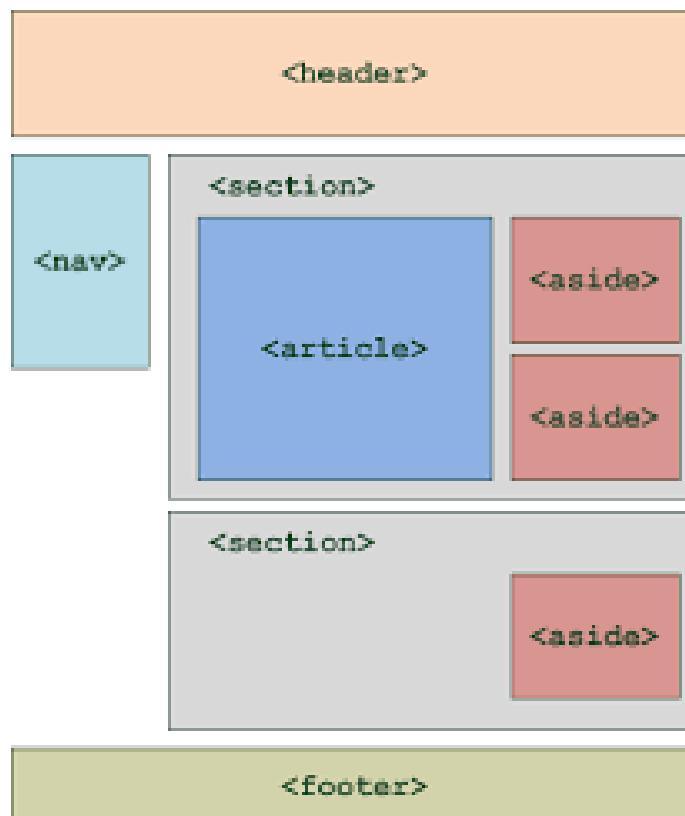
À partir de votre **Maquette**, listez toutes les **balises HTML** que vous allez utiliser.

Exemple :

- En-tête → `<header>`, `<nav>`, ``, ``, `<a>`
- Contenu → `<main>`, `<section>`, `<h2>`, `<p>`, ``
- Formulaire → `<form>`, `<input>`, `<label>`, `<button>`
- Pied de page → `<footer>`

Le but est de **penser en HTML avant de coder**.

Exemple:



Challenge 2 — Créer une clé SSH + dépôt Git complet

Vous allez mettre en place une **clé SSH** et configurer un dépôt Git pour votre projet.

Ce guide suit les **documentation Git 2025** (versions récentes).

Étapes communes

1. Générer une clé SSH (ed25519).
2. Ajouter la clé privée à l'agent SSH.
3. Copier la clé publique dans **GitHub/GitLab → Settings → SSH Keys**.
4. Créer un dépôt Git (vide) sur GitHub/GitLab.
5. Initialiser votre projet localement et créer un `README.md`.
6. Ajouter une capture d'écran de votre page dans le dépôt.
7. Faire un commit et **pousser (push)** sur le dépôt distant.

Guide pour la création d'une clé SSH:

[Introduction - Clé SSH et Git pour les débutants.pdf](#)

Windows (Git Bash / PowerShell)

1. Génération de la clé

```
ssh-keygen -t ed25519 -C "votre_email@example.com"
```

2. Démarrage de l'agent SSH

```
eval "$(ssh-agent -s)"
```

3. Ajout de la clé privée

```
ssh-add ~/.ssh/id_ed25519
```

4. Afficher la clé publique

```
cat ~/.ssh/id_ed25519.pub
```

→ copier dans GitHub / GitLab (SSH Keys)

5. Configuration Git

```
git config --global user.name "Votre Nom"
```

```
git config --global user.email "votre_email@example.com"
```

6. Initialiser projet

```
git init
```

```
echo "# Maquette Projet" > README.md
```

7. Ajouter capture

Exemple : maquette.png

```
git add README.md maquette.png
```

```
git commit -m "feat: ajout README et capture de la maquette"
```

8. Lier dépôt distant

```
git remote add origin git@github.com:VotreNom/maquette.git
```

9. Push vers GitHub

```
git push -u origin main
```

Pop!_OS (Linux)

Terminal

```
ssh-keygen -t ed25519 -C "votre_email@example.com"
```

```
eval "$(ssh-agent -s)"
```

```
ssh-add ~/.ssh/id_ed25519
```

```
cat ~/.ssh/id_ed25519.pub # copier clé publique
```

```
git config --global user.name "Votre Nom"
```

```
git config --global user.email "votre_email@example.com"
```

```
git init
```

```
echo "# Maquette Projet" > README.md
```

```
git add README.md capture.png
```

```
git commit -m "feat: ajout README et capture"
```

```
git remote add origin git@github.com:VotreNom/maquette.git
```

```
git push -u origin main
```

macOS (Terminal)

Terminal

```
ssh-keygen -t ed25519 -C "votre_email@example.com"
```

```
eval "$(ssh-agent -s)"
```

```
ssh-add --apple-use-keychain ~/.ssh/id_ed25519
```

```
pbcopy < ~/.ssh/id_ed25519.pub # copie automatique
```

```
git config --global user.name "Votre Nom"
```

```
git config --global user.email "votre_email@example.com"
```

```
git init
```

```
echo "# Maquette Projet" > README.md
```

```
git add README.md capture.png
```

```
git commit -m "feat: ajout README et capture"
```

```
git remote add origin git@github.com:VotreNom/maquette.git
```

```
git push -u origin main
```