

Features	User Stories	Pondération	Commentaires
Template principal	Créer le template commun à toutes les "pages" (ou vues) : un header, un footer, une nav et un bloc dynamique Jinja (mkdir templates, code layout.html)	1	
Base de données	Créer le fichier de base de données pour gérer la sauvegarde des données utilisateurs : fichier init_db.py	1	
Template téléchargement d'images	Créer le template (form.html) de téléchargement en exportant le template principal et en modifiant le block dynamique (templates/form.html) Dans le bloc dynamique, créer un formulaire pour récupérer les données de l'utilisateur. Le formulaire contient 4 champs (input) : image, titre, description, catégorie	4	
Template vue détails	Créer le template (details.html) pour afficher les détails d'une image en exportant le template principal et en modifiant le block dynamique (templates/detail.html). Dans le bloc dynamique, afficher 3 éléments : image, titre, description.	3	
Page d'accueil	<ul style="list-style-type: none"> <li>- affiche les images de la base de données par date de création et par ordre décroissant</li> <li>- barre latérale avec liens vers chaque catégorie</li> <li>- filtre par catégorie en fonction des paramètres de requête</li> <li>- Cliquer sur une image mène à une page qui affiche l'image avec son titre et sa description</li> <li>- compléter le bloc dynamique du template principal (layout.html) pour afficher les prérequis</li> </ul>	3	<ul style="list-style-type: none"> <li>- Un en-tête avec le nom du site, un lien vers la page d'accueil et un lien vers la page de téléchargement</li> <li>- Un pied de page avec une ligne de copyright</li> <li>- Une barre latérale avec des liens pour filtrer les images par catégorie</li> </ul>
Fichier <a href="#">app.py</a>	Créer un fichier <a href="#">app.py</a> qui contiendra toutes les routes de notre application et la génération de la base de données.	2	
Implémenter les routes	Implémenter toutes les routes dans le fichier <a href="#">app.py</a> . Les routes de toutes les vues (form, details, etc...), les routes vers les dossiers images... Les routes en GET, en POST selon les cas.	5	
Validation des formulaires	Ajouter une vérification dans le fichier app.py pour vérifier le format des images importées dans la base de données. Utiliser la fonction <code>allowed_file()</code> (voir doc Flask)	1	
Commentaires	<ul style="list-style-type: none"> <li>- Les utilisateurs peuvent ajouter un commentaire sur la page de présentation d'images</li> <li>- Tous les commentaires pour une image donnée sont affichés sur la page d'affichage d'image</li> <li>- ajouter cette fonctionnalité dans le template (details.html)</li> </ul>	5	
Mise en commun et assemblage du code	Imbrication de toutes les parties du code de l'équipe. Tester les routes, les sauvegardes en base, les fonctionnalités.	5	
Design CSS/HTML	Optimisation CSS et présentation propre de l'application globale.	5	