

MonDressing									
Tableaux									
Personnel									
Privé									
Inviter									
Afficher le menu									
A faire Admin	À faire Front	A faire Back	En cours	Terminé front	Terminé back	V2	Terminé admin	+ Ajouter une autre liste	
Faire la maquette d'une page de mon appli : page d'accueil	Front - page d'accueil V1 : Bienvenue "Nom de l'utilisateur" + bouton météo-bouton création vêtement	Back - définir les requêtes sql permettant de retourner les informations nécessaires à la modification : la suppression des catégories, des marques, des couleurs, des caractéristiques et des occasions	Tous les écrans sont responsive	Sécurité : token (jeton d'authentification) lors de l'authentification de l'utilisateur.	Back - définir les requêtes sql permettant de retourner les informations nécessaires à la suppression d'un vêtement	Front - intégrer un widget météo ou il est possible de choisir sa ville	Détailier mes user stories + spécifications techniques		
Monter mon dossier projet	Front - hasher le mot de passe AVANT de l'envoyer au back avec librairie Bcrypt	Back - harmoniser jeu de données	Back - faire la requête permettant de comparer un mot de passe hashé récupéré d'un formulaire avec le mot d epasse enregistré en base	Front - ajouter une validation lors de la suppression : La page doit demander la confirmation avant suppression définitive : au click sur la poubelle, on demande si la personne est sure, si oui, on supprime, si non, on revient à la modale sans rien supprimer.	Sécurité - requêtes sql préparées évitant les injections SQL	Ajouter les logo des marques en BDD et afficher ça plutôt que le nom de la marque dans les pages de visionnage.	Faire le dictionnaire des données		
Monter mon dossier professionnel	Front - envoyer avec chacune des requêtes faites au back un header avec le token JWT transmis par le back lors de la connexion, contenant l'id de l'utilisateur, afin que l'utilisateur n'ai accès qu'à ses vêtements à lui et pas ceux de toute la BDD	Faire des tests unitaire : couvrir le max de l'application	Front - login + register - supprimer les : et échapper les ' avant envoi au back	Front - intégrer un widget météo	Back - définir les requêtes sql et fonctions permettant de retourner la liste des noms de vêtements, des noms des couleurs, catégories, marques, caractéristiques, occasions, pour vérifier l'unicité côté front	Stocker en base le code hexadécimal de chaque couleur pour l'afficher à coté dans les pages de visionnage, création et modification.	Faire mon cahier des charges		
Veille en anglais : recherche concernant un problème technique ou nouvelle fonctionnalité: décrire besoin d'information (pb rencontré), préciser comment a été effectuée la recherche (mots clés, liste des sites retournés), critère de sélection du ou des sites choisis, indiquer la solution trouvée, et si elle a pu être mise en oeuvre + + extrait du site anglophone utilisé dans la recherche accompagné de la traduction en français - 750 signes	Front - ajouter des vérifications sur les formulaires d'enregistrement et de connexion : @, password chiffres lettres, etc	Back - implémenter les certificats https	+ Ajouter une autre carte	Front - Faire la page permettant de s'enregistrer	Back - faire les requetes sql permettant d'enregistrer une nouvelle marque, couleur, caractéristique, occasion, catégorie dans le formulaire de création d'un vêtement	Afficher à l'utilisateur des cases permettant de choisir la couleur et qui me renverront le code hexadécimal à associer : sur les pages de visionnage, création, modification.	Faire un glossaire		
Support de présentation Prezzi ou PowerPoint	Front - ajout fonctionnalité reset password	Back - supprimer les : et échapper les ' des login et mdp reçus du front		Front - Faire la page permettant de se connecter	Back - faire les requêtes sql permettant de retourner les informations nécessaires à la création d'un vêtement	Une fois que l'on a saisi assez de données indiquant ce qui va ensemble etc : mettre de l'IA pour déterminer ce qui va ensemble et ne va pas ensemble	Maquetter mon application : mockup fil de fer de mes écrans, faire une planche présentant les couleurs et écritures		
+ Ajouter une autre carte	Front - Faire en sorte que l'ID de l'utilisateur soit envoyé dans chaque token pour que l'utilisateur n'ait accès qu'à ses vêtements à lui et non pas tous ceux de la BDD	Back - ajout requête permettant la modification du mot de passe		Front - Faire la page permettant de s'enregistrer	Back - définir les requêtes sql permettant de retourner les informations nécessaires à l'affichage	Ajouter en base dans la table catégorie le lien vers le font-awesome pour ajouter la tuile correspondante.	Faire le résumé du projet en 200, 250 mots		
	Front - Mettre du dynamisme dans tous mes formulaires : vérification d'unicité, désactivation du bouton tant que les champs ne sont pas tous	Back - decrypter le token JWT à chaque envoi du front pour en sortir l'ID_USER à mettre dans chacune des requêtes (fonction jwt.decode()), afin que l'utilisateur n'ai accès qu'à ses vêtements à lui et pas ceux de toute la BDD		Front - Implémenter logique gestion login : création d'un compte, et connexion	Back - utiliser le bon code HTTP pour retour quand le vêtement n'existe pas en base de données (code 204)	Faire la vérification sur les selects des formulaires : vérifier que les marques, couleurs, etc n'existent pas déjà en base de données (majuscules, pas de majuscules, etc)	Veille sécurité : recherche sur les injections sql etc : sites, mots clés utilisés, vulnérabilités trouvées, failles potentielles corrigées		
	+ Ajouter une autre carte	Back - ajouter un salt à mon cryptage de password		Angular - créer un composent en cas d'erreur : dans app-routing.ts : {path: '', component : LoginComponent}, -> en cas d'erreur, l'affichage est	Back - définir les requêtes sql permettant de retourner les informations nécessaires à l'affichage	+ Ajouter une autre carte	+ Ajouter une autre carte		
		+ Ajouter une autre carte		+ Ajouter une autre carte					