

A. INTRODUCTION A OSMOZ		NOTES
Contexte et présentation du projet		Cf : CdC / Objectif
B. COMPETENCES DU REFERENCIEL		
Activité type 1 – Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité		
† CP 1 – Maquetter une application		Création des wireframes sur Whimsicale – User stories en annexe markdown
† CP 2 – Réaliser une interface utilisateur web statique et adaptable		Création d'une page Contact, intégrant un formulaire. Responsive de mobile à grand écran.
† CP 3 – Développer une interface utilisateur web dynamique		Création d'une page d'accueil bouclant dynamiquement sur les produits récupérés via l'API
Activité type 2 – Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité		
† CP 5 – Créer une base de données		Réalisation du MLD, MCD, Dictionnaire de données
† CP 6 – Développer les composants d'accès aux données		Gérer la connection et le stockage des informations de connexion (cookies)
† CP 7 – Développer la partie back-end d'une application web ou web mobile		Mise en place d'un back office permettant la mise à jour des données utilisateurs / produits
C. Résumé du projet Osmoz		Cf : CdC / Objectif + Déroulement du projet

D. Cahier des charges		
I - La conceptualisation de l'application		
† Le Minimum Viable Product		Cf : CdC / Fonctionnalités
† Les fonctionnalités du projet		Cf : CdC / Fonctionnalités
† Le Workflow de l'application		Gestion des user stories via Trello
II - La mise en dimension de l'application		
† Les wireframes de l'application		Réalisés sur Whimsicale(version utilisateur connecté)
† La création de la charte graphique		Documentation sur la psychologie des couleurs. Suivi des tendances UI UX
E. Spécifications techniques de Osmoz		
I – Le versionning		
		Gestion du projet sur 2 repos distincts (back/front). Chaque membre développait sur sa branche, avec merge tous les soirs. Revu des conflits chaque matin après récupération d'une base commune.
II – Les technologies Back-End		Cf : CdC / Technologies
III – Les technologies Front-End		Cf : CdC / Technologies
IV - Sécurité		Cors / sanitizing des données d'input / token

## E. Organisation de l'équipe

I – Présentation de l'équipe	Cf : CdC / L'équipe
II – Organisation du travail	Journée de 9h à 17h. Revu des conflits de version, stand-up meeting assis. Communication régulière durant la journée via Discord+Slack.
III – Outils utilisés	Discord / Slack / Trello / VSC PairPro
IV – Programme des sprints	Cf : Journal de bord
G. Production	
Réalisations Personnelles :	
† Sprint 0 – Réalisation et rédaction du cahier des charges† Sprint 0 – Réalisation et rédaction du cahier des charges	Cf : CdC + carnet de bord
† Sprint 1 – Mise en place de l'architecture React-Redux	Carnet de bord
† Sprint 2 – Implantation d'Axios + Connexion Front/Back	Carnet de bord
† Sprint 3 – Développement de la messagerie + Dynamisation de composants	Carnet de bord
† Sprint 4 – Résolution des bugs / mise en prod Front	Carnet de bord
V – Jeu d'essai	
† Beta Test	Proposer du beta testing à des tiers, afin de repérer les erreurs d'UI - UX
† Test unitaire	Utilisation de Jest-DOM

## F. Organisation de l'équipe

I – Présentation de l'équipe	Cf : CdC / L'équipe
II – Organisation du travail	Journée de 9h à 17h. Revu des conflits de version, stand-up meeting assis. Communication régulière durant la journée via Discord+Slack.
III – Outils utilisés	Discord / Slack / Trello / VSC PairPro
IV – Programme des sprints	Cf : Journal de bord
G. Production	
Réalisations Personnelles :	
† Sprint 0 – Réalisation et rédaction du cahier des charges† Sprint 0 – Réalisation et rédaction du cahier des charges	Cf : CdC + carnet de bord
† Sprint 1 – Mise en place de l'architecture React-Redux	Carnet de bord
† Sprint 2 – Implantation d'Axios + Connexion Front/Back	Carnet de bord
† Sprint 3 – Développement de la messagerie + Dynamisation de composants	Carnet de bord
† Sprint 4 – Résolution des bugs / mise en prod Front	Carnet de bord

## G. Troubleshooting

Problématique	Solution trouvée
† Utilisation d'images dans le code JSX	Nous avons fini par réaliser qu'il fallait utiliser l'image sous forme de Composent, pour matcher avec React.
† Redémarrer automatiquement une session via token	En complément du token, le localStorage garde également l'id de l'utilisateur courant, afin de récupérer les informations même après avoir fermé le navigateur
† Connexion Front/Back avec server Back AWS	Nous avons tenté de trouver une solution nous même, mais avons fini par recourir à Nelmio (bundle)
† Transmettre des infos lors d'une requete GET	Utilisation d'une requete POST pour transmettre un body
H. Conclusion	
I – Développement V2	
† La modération de contenu	Ajout d'un rôle Modérateur, permettant la suppression/modification de produits
† Validation	Modérateurs et Admins doivent valider un produit avant sa diffusion
† Messagerie instantannée	Installation de WebSocket pour créer un abonnement à l'action « Send Message »

## H. Annexes

WIREFRAMES	Dictionnaires des données
MCD	Crédits
MLD	