

☒ Cahier des charges – Application BURNOUT

1. Présentation Générale

1.1. Nom du projet

BURNOUT

1.2. Objectif

Créer une WebApp sociale dédiée aux motards permettant de :

- Partager des photos de motos, roadtrips et balades.
- Publier ou découvrir des roadbooks (fichiers GPX ou tracés personnalisés).
- Échanger par messagerie instantanée.
- Suivre d'autres utilisateurs, liker/commenter les publications.
- Utiliser des hashtags pour classer et explorer les contenus.
- Consulter des stories temporaires.

1.3. Public cible

Motards et passionnés de deux-roues.

2. Design et Interface

2.1. Design

- Thème sombre par défaut (InertiaJS + Vue).
- Style sobre et moderne, proche d'Instagram.
- **Code couleur** : Noir / Orange / Blanc.
- Fonts modernes, contrastes marqués.

2.2. Responsiveness

- **Mobile** : Menu bottom tab (Feed / Roadbooks / Messages / Profil).
- **Desktop** : Sidebar verticale à gauche.

2.3. Maquettes

- Page de connexion déjà intégrée (style clair).
 - Maquette à venir pour le feed principal avec :
 - Barre de navigation
 - Affichage des stories
 - Affichage des posts (image / légende / likes / commentaires)
-

3. Fonctionnalités

3.1. Authentification

- Email + mot de passe (Laravel)
- Connexion Google (OAuth2)
- Récupération du mot de passe par email

3.2. Profil Utilisateur

- Avatar
- Biographie
- Nombre de followers / followings
- Page de profil publique
- Edition de profil personnelle uniquement

3.3. Feed d'actualités (Posts)

- Photo obligatoire + Légende optionnelle (avec hashtags)
- Hashtags cliquables → page dédiée
- Scroll infini
- Likes & commentaires
- Tri chronologique inverse

- Pagination backend (cursor-based)

3.4. Stories

- Affichage en bandeau en haut du feed
- Image uniquement
- Durée de vie : 24h
- Suppression automatique via cron

3.5. Roadbooks

Création :

- Upload de fichiers GPX ou KML
- Ou saisie manuelle via carte (Leaflet)
- Titre + description obligatoires

Affichage :

- Image de carte avec tracé
- Infos du créateur
- Description
- Feed de roadbooks dédié
- Téléchargement du fichier d'origine

3.6. Messagerie instantanée

- 1-1 uniquement
- Stockage des messages
- Notifications en temps réel via Pusher
- Badge non-lu sur message reçu
- Historique accessible (scroll)

3.7. Notifications

Événements notifiés :

- Nouveau like sur un post
- Nouveau commentaire
- Nouveau message
- Nouveau follow

Caractéristiques :

- Temps réel via Pusher
- Stockées en base
- Badge de lecture
- Historique limité à 7 jours (page dédiée)

3.8. Hashtags

- Utilisables uniquement dans les légendes des posts
- Cliquables → page listant les posts avec ce tag
- Stockés dans une table dédiée (hashtags , post_hashtag)

4. Langues

- Français & Anglais
- Détection automatique + option de changement via le menu

5. Infrastructure

5.1. Stack technique

- **Backend** : Laravel 10, PHP 8.2
- **Frontend** : InertiaJS + Vue 3
- **Base de données** : MySQL
- **Authentification** : Laravel Breeze + Socialite
- **WebSocket** : Pusher
- **Email** : SMTP

Stockage images :

- Utiliser **Cloudinary** (formule gratuite recommandée)
- Alternatives : Imgix, Uploadcare, stockage S3-compatible

5.2. Hébergement

- **Premium Hostinger**
- Stockage local limité (80GB) → éviter stockage direct des images
- Cron jobs disponibles pour suppression automatique des stories

6. Architecture des données (extrait simplifié)

6.1. Tables principales

- users
- posts
- comments
- likes
- follows
- stories
- hashtags
- post_hashtag
- messages
- roadbooks
- notifications

7. Découpage MVP

Niveau	Fonctionnalité	Objectif
MVP 1	Auth + Profil + Feed + Follow	Base de réseau social
MVP 2	Likes + Commentaires + Hashtags	Engagement
MVP 3	Roadbooks (GPX) + Feed dédié	Contenu moto spécifique
MVP 4	Messagerie + Notifications temps réel	Interaction
MVP 5	Stories + UI avancée + Mobile polish	Expérience utilisateur

8. Contraintes & Limites techniques

- Pas de NodeJS sur l'hébergement → build frontend local puis déploiement des assets.
- Pusher obligatoire pour WebSockets.
- Stockage cloud recommandé (Cloudinary).
- Compression des images avant upload.
- Pagination pour tous les feeds.

9. Sécurité

- Protection CSRF
- Filtrage XSS dans commentaires et descriptions
- Rate limiting sur endpoints sensibles (login, post, etc.)
- Authentification Google sécurisée via Socialite

10. Évolutions futures envisagées

- Stories vidéos

- Groupes de discussions
- Roadbooks collaboratifs
- Modération communautaire