

# Cahier des charges

---

## 1. Aperçu de projet

### 1.0. Nom du projet : Click&Rent

### 1.1. Description

Click&Rent est une plateforme web de gestion de prêt de matériel. L'objectif est de développer un site web en PHP utilisant le modèle MVC (Modèle-Vue-Contrôleur) pour permettre aux utilisateurs de consulter le matériel disponible, faire des demandes de prêt, et aux administrateurs de gérer les prêts

### 1.2. Objectifs

- Développer un site web en PHP suivant le modèle MVC
- Implémenter un système de gestion de prêt de matériel
- Intégrer une base de données MariaDB pour le stockage des données
- Créer une interface utilisateur intuitive pour les différentes fonctionnalités

### 1.3. Public cible (le persona est dans 3.2)

- Utilisateurs souhaitant emprunter du matériel
- Administrateurs gérant les prêts et le matériel

## 2. Aperçu du site web

### 2.1. Informations

- **Nom du site web** : Click&Rent
- **Type de site web** : Plateforme de gestion de prêt de matériel

### 2.2. Fonctionnalités clés principales :

#### Pour les utilisateurs :

- **Authentification** : Permet à l'utilisateur de s'identifier et se connecter à son espace.
  - Inscription et connexion sécurisée.
- **Consultation du catalogue** : Liste des matériels disponibles.
  - Consultation des articles disponibles avec description, état et photos.
  - Fonctionnalité de recherche et filtres avancés.

- **Demande de prêt** : L'utilisateur pourra faire une demande de prêt.
  - Formulaire de réservation avec sélection du matériel, dates de début et fin.
  - Suivi des demandes en cours et historique des prêts.
- **Suivi des prêts** : Historique des demandes et statut des prêts en cours.
- **Retour du matériel** : Signalement du retour via l'interface utilisateur.

#### Pour les administrateurs :

- **Gestion du matériel** : Ajout, modification et suppression d'articles dans le catalogue.
- **Validation des demandes** : Interface pour accepter ou refuser les demandes de prêt.
- **Suivi des prêts** : Tableau de bord pour visualiser les prêts actifs et l'historique.
- **Gestion des retours** : Vérification et validation des retours.

### 2.3. Technologies utilisé :

- **Frontend** : HTML, CSS, JavaScript
- **Backend** : PHP (modèle MVC)
- **Base de données** : MariaDB
- **Contrôle de version** : Git avec GitHub pour la collaboration

## 3. Design et User Experiences (UX)

### 3.1 Design de conception

- Interface moderne et conviviale adaptée à tous les appareils (responsive design).
- Navigation intuitive avec une catégorisation claire du matériel.
- Images haute qualité accompagnées d'une description détaillée.

### 3.2 Persona d'utilisateur

Thomas Dupont

Enseignant en informatique

#### SITUATION

Professeur dynamique et organisé

Thomas a besoin d'emprunter régulièrement du matériel pour ses cours pratiques.

#### PROFIL

Genre : Homme

Age : 35 ans

Situation professionnelle : Enseignant titulaire

#### OBJECTIFS

Thomas souhaite pouvoir réserver facilement du matériel à l'avance pour préparer ses cours sans avoir à se déplacer physiquement au service de prêt.

## INTÉRÊTS

Ayant un emploi du temps chargé, Thomas apprécie les interfaces simples et efficaces. Il souhaite pouvoir visualiser rapidement la disponibilité du matériel et suivre l'état de ses demandes.

## PRÉOCCUPATIONS

Thomas a besoin d'un système fiable qui lui confirme ses réservations et lui rappelle les dates de retour.

### 3.3. Flow de l'utilisateur

ex : Accueil → Connexion → Catalogue de matériel → Détail du matériel → Demande de prêt → Confirmation → Suivi des prêts

### 3.4. Information d'architecture

#### 3.4.1. Structure du site

- **Accueil ok**
- Boutique de matériel ok
- Mon compte
  - **Connexion ok**
  - **Inscription ok**
  - Tableau de bord ok
    - Historique des prêts
    - Demandes en cours
- Demande de prêt (Panier) ok
- Espace administrateur

## 3.5 Mockup et Prototypes

### 3.5.1. Page d'accueil



### 3.5.2. Page boutique de matériel

[img]

### 3.5.3. Page tableau de bord utilisateur



John DOE

ME DECONNECTER

Mes informations

Historique des prêts

Demandes en cours

Historique de vos locations

Nom du produit	Date de début	Date de fin	Statut
 Ryzen 7800x3d	15/06/2025	18/07/2025	Terminé



John DOE

ME DECONNECTER

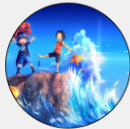
Mes informations

Historique des prêts

Demandes en cours

DEMANDES EN COURS

Nom du produit	Date de début	Date de fin	Statut
 Ryzen 7800x3d	15/06/2025	18/07/2025	Accepté
 RTX 4090 Suprim X	15/06/2025	19/08/2025	En attente

Nos matériels 

John DOE

ME DECONNECTER

Mes informations

Historique des prêts

Demandes en cours

## COORDONEES

Nom :

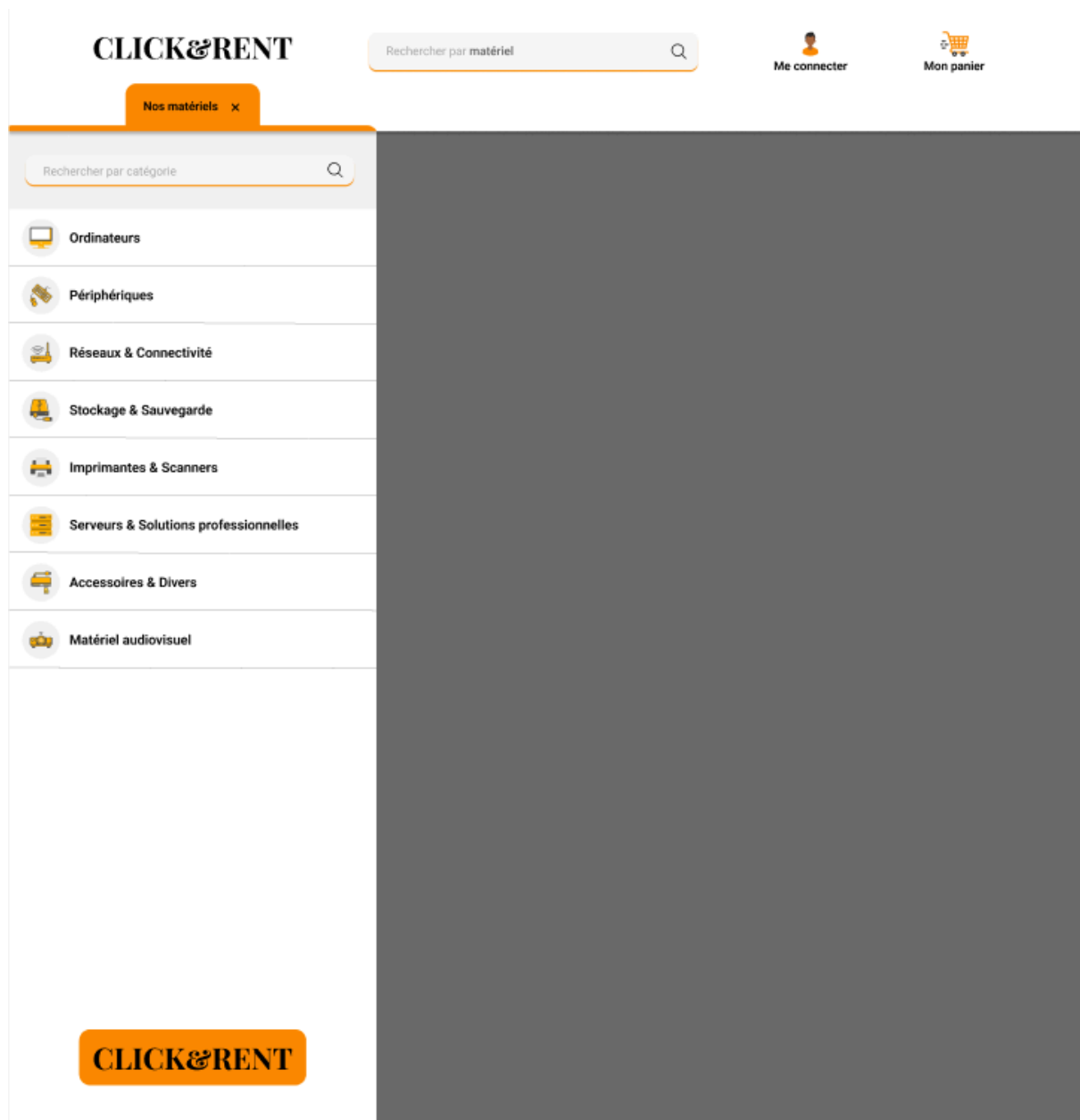
Prénom :

Email :

Valider



### 3.5.4. Menu déroulant



### 3.5.5. Page de connexion

# CLICK&RENT



 **Pour gérer vos locations,**  
connectez-vous ou créez votre compte.

**Se connecter**

**Vous n'avez pas encore de compte !** [Créer votre compte !](#)

### 3.5.6. Page d'inscription



# CLICK&RENT



 **Louez en toute simplicité !**  
et créez votre compte

**Créer un compte**

**Vous possédez déjà un compte !** [Se connecter !](#)

## 3.6 Guide de style

### 3.6.1. Palette de couleur

- Primaire : **Orange** (#FC8B00)
- Secondaire : **Rose** (#FA2ABE)
- Accentuation : **Violet** (#A91CF0)
- Background : Blanc (#FFFFFF)
- Texte : Noir (#212121), Gris (#757575)

### 3.6.2. Typographie

- Titres : Montserrat
- Texte de body : Open Sans
- Texte de boutons : Roboto

## 4. Développement

### 4.1 Architecture MVC

Le site sera développé en suivant le modèle MVC (Modèle-Vue-Contrôleur) pour garantir une séparation claire entre la logique métier, l'affichage et le contrôle.

- **Modèle** : Gère la logique métier et les données, fournit une interface simple au contrôleur avec des algorithmes et requêtes SQL
  - Classes PHP pour les entités (Utilisateur, Matériel, Prêt) et gestion de la logique métier
- **Vue** : S'occupe de l'affichage en utilisant principalement du HTML, avec des structures de contrôle PHP pour présenter les données
  - Templates HTML avec CSS pour l'affichage des informations
- **Contrôleur** : Interagit avec l'utilisateur, dirige les actions sur le modèle et transmet les résultats à la vue pour l'affichage final
  - Gestion de la navigation et interaction entre modèle et vue

### 4.2. Base de données

- Utilisation de MariaDB
- Tables principales :
  - **Utilisateurs** : ID, nom, prénom, email, mot de passe, rôle (utilisateur/admin).
  - **Matériel** : ID, nom, description, état, photo, disponibilité.
  - **Prêts** : ID prêt, ID utilisateur, ID matériel, date début, date fin, statut.

### 4.3. Optimisation pour les moteurs de recherche (SEO)

- Structure de l'URL optimisée pour chaque page de produit.
- Utilisation de balises meta-descriptions et de titres pertinents.
- Création d'un sitemap XML pour faciliter l'indexation.

## 5. Gestion de projet

### 5.1 Rôles et responsabilités

- Développeurs backend
- Développeurs frontend
- Designer UX/UI

## 6. Livrables

- Code source complet du site web.
- Script SQL pour la création et le peuplement de la base de données.
- Documentation détaillée incluant l'architecture technique, les fonctionnalités et un guide utilisateur.

## 7. Appendix

- Figma :  
<https://www.figma.com/design/ZXTYy00Jyf3MQqRZqwGNR6/Projet-de-location-de-mat%C3%A9riel?node-id=0-1&p=f&t=fdSoDnCPcixV1zYy-0>