

# CAHIER DES CHARGES SITE FABLAB ROBOTIQUE

Obed - Mathis - Alexis



AJC Formation

## 1. Présentation du Projet

Le présent projet de **FabLab Robotique** a pour objet d'établir un site web opérationnel et complet de nature à permettre l'édition d'un site web dont la présentation des activités, des projets, des articles de Fablab soit facilitée.

Le site se décline en deux parties :

la partie publique visible de tous les visiteurs permettant d'accéder aux projets, articles publiés, ainsi que des vidéos de la webTV consultables et à la découverte de Fablab, et

la partie administration (dashboard) réservée à la zone des utilisateurs (Admin) autorisés. Cette zone étant surtout à disposition pour gérer le contenu de l'édifice du site web au travers des fonctionnalités d'ajouts de mise à jour ou suppression des projets articles ou vidéos WebTV.

D'où l'objectif principal reste de réaliser une plate-forme dont l'utilisation reste simple, moderne, sécurisée et bien agencée afin que le client au final puisse sans aucune connaissance technique gérer son site.

Le site repose sur une structure **d'architecture MVC**(Modèle, Vue, Contrôle) en vue du maintien d'un code propre, structuré, et d'une MySQL afin de stocker l'ensemble de la base de données en prévoyant plusieurs protections telles que la vérification des rôles, le filtrage pour les champs, **la protection SQL**(Strutured query language), et la gestion sécurisée des mots de passes.

Aussi ce projet assure au Fablab une présence sur la toile claire professionnelle et aisément mise à jour pour tous les visiteurs de navigation fluide.

## 2. Objectifs du Projet

Le projet **Fablab Robotique** a plusieurs objectifs principaux qui visent à offrir un site web complet, moderne et facile à utiliser pour le client comme pour les visiteurs.

### 1 - Centraliser toutes les informations du Fablab

Le site doit regrouper au même endroit : les projets réalisés par les étudiants ou le Fablab, les articles publiés sur les activités, formations ou événements, les vidéos de la WebTV interne, les informations importantes sur le fonctionnement du Fablab. L'objectif est que tout soit accessible rapidement, sans passer par plusieurs plateformes différentes.

L'objectif est que tout soit **accessible rapidement**, sans passer par plusieurs plateformes différentes.

### 2 - Faciliter la gestion du contenu

Le client doit pouvoir gérer lui-même, sans difficulté : les projets (ajout / modification / suppression), les articles, les vidéos WebTV, les comptes utilisateurs. Grâce au dashboard administrateur, la gestion doit être : simple, rapide, sécurisée, utilisable par des personnes non techniques.

### 3 - Proposer une expérience utilisateur de qualité

Le site Internet doit présenter une interface à la fois plaisante, intuitive et adaptée aux attentes des utilisateurs : présentation claire des projets, images facilement visibles, systèmes de filtres et de recherche pour un confort de temps, mise en avant des contenus récents, pages structurées et très lisibles. L'objectif est de donner au site un aspect professionnel et engageant.

### 4 - Assurer que la structure technique est propre et évolutive

Le site doit pouvoir être assez facilement maintenu et évolué à l'avenir : architecture MVC propre, code propre, sécurisé et bien organisé, base de données MySQL bien structurée, système de droits (Admin / Éditeur / Utilisateur), possibilité d'ajouter d'autres sections ultérieurement.

### 5 - Offrir un espace sécurisé pour l'administration

Le dashboard (page administration) doit garantir : la protection des données, le contrôle des accès, le stockage sécurisé des informations, la vérification des connexions utilisateurs, le hachage (chiffrement mots de passe dans la base de données SQL) des mots de passe. L'objectif est d'éviter toute modification non autorisée et de protéger le site.

## **6 - Mettre en avant les compétences du Fablab**

En présentant :les projets techniques, les réalisations des étudiants, les activités du Fablab, les vidéos et démonstrations, le site doit permettre de valoriser le Fablab auprès :des nouveaux étudiants, des partenaires, des entreprises, du public.

# 3. Les besoins Fonctionnels

(ce que le site doit permettre de faire nécessairement – les fonctionnalités visibles de l'utilisateur ou de l'administrateur)

## A. Fonctionnalités visibles des visiteurs (Front-Office)

### 1. Consulter les projets

Voir une liste de tous les projets récents.

Voir l'image du projet (image locale ou URL).

Ouvrir la fiche projet détaillée (description, technologies, images, ...).

Rechercher un projet via une barre recherche.

Filtrer par catégorie (Robotique / Drone / Autres).

### 2. Lire les articles

Voir les articles avec image, titre, extrait et date.

Lire un article complet.

Afficher l'image de l'article en grand dans le header de la page.

Recherche d'articles via une barre dédiée.

### 3. Accéder à la Web TV

Lire une vidéo YouTube intégrée.

Voir une liste de vidéos recommandées.

Naviguer entre les vidéos sans recharger toute la page.

Lire ou ajouter des commentaires (pour les visiteurs anonymes si activé).

### 4. Navigation

Accès rapide aux pages via un header clair.

Footer avec informations / liens / contact.

## **2. Lire les articles Voir les articles en image, titre, extrait, date.**

Lire l'article complet.

Affichage de l'image de l'article en grand dans le header de la page.

Recherche d'articles via une barre dédiée.

## **3. Accéder à la WebTV**

Visionner un vidéo Youtube intégrée.

Visionner les vidéos de la WebTV (liste de vidéos recommandées).

Naviguer entre les vidéos qui sont visibles sans nécessiter un rechargement de toute la page.

Ajouter un commentaire ou le lire (visiteur anonyme si activé).

## **4. Navigation**

Accès rapide aux pages du site grâce à header clair.

footer avec informations / liens / contact.

# **B. Fonctionnalités Administrateurs & Éditeurs (Back-Office / Dashboard)**

## **1. Gestion des utilisateurs**

Connexion et déconnexion sécurisée.

Création de comptes avec rôles : Admin / Éditeur / Utilisateur.

L'Admin peut gérer les rôles.

## **2. Gestion des articles**

Créer un article.

Modifier un article.

Supprimer un article.

Ajouter une image : depuis un lien internet (URL), ou depuis une image locale déjà connue / uploadée.

### **3. Gestion des projets**

Consulter les des projets.

Ajouter un nouveau projet.

Modifier un projet existant.

Supprimer un projet.

Ajouter une image locale ou une URL.

Ajouter les technologies utilisées (séparées par virgules).

Catégorisation automatique (robotique / drone / autres).

### **4. Gestion WebTV**

Ajouter une nouvelle vidéo (ID YouTube).

Modifier une vidéo existante.

Supprimer une vidéo.

Gérer tous les commentaires associés.

## **C. Fonctionnalités système (automatisations & logique)**

### **1. Catégorisation automatique**

Le système détecte automatiquement des mots-clés pour ranger le projet :

ROBOT → robotique

DRONE FPV RACING → drone

LE RESTE → autres

### **2. Images intelligentes**

Si l'image ne charge pas → icône alternative affichée.

Si l'image est une URL → affichage direct.

Si l'image est locale → elle apparaît via public/images/projets/.

### **3. Sécurité**

Empêcher l'accès au dashboard sans être connecté.

Empêcher l'accès si rôle ≠ Admin / Éditeur.

Hashage des mots de passe.

Vérification des formulaires (protection basique contre injection).

## **D. Accessibilité & ergonomie**

Site responsive (PC / tablette / mobile).

Police lisible (Inter).

Boutons visibles, contrastés.

Images adaptées selon taille d'écran.

## **E. Structure technique attendue**

Architecture MVC simple (Model / View / Controller).

Base de données MySQL.

Code PHP organisé dans app/.

Fichiers CSS dans "/public/css/".

Système de routing simple via index.php.

# 4. Besoins Non-Fonctionnels

Tout ce qui concerne la qualité, les performances, la sécurité, l'ergonomie, la structure technique

## A. Performance & Rapidité

### 1. Temps de chargement réduit

- Les pages doivent se charger en **moins de 2 secondes**.
- Les images sont optimisées (formats : .jpg, .png, .webp, rien d'autres ! ) Empêcher l'extraction si autres formats .
- Le code CSS/JS doit être léger et organisé.

### 2. Structure MVC efficace

Séparer les responsabilités :

**Modèles** : accès base de données

**Contrôleurs** : logique métier

**Vues** : affichage / interface

Cela facilite la maintenance et les modifications.

### 3. Utilisation minimale de requêtes MySQL

Réduire les requêtes trop répétitives.

Utilisation de prepare() + fetchAll() correctement.

## B. Sécurité

### 1. Authentification sécurisée

Hashage des mots de passe avec password\_hash().

Vérification via password\_verify().

### 2. Accès restreints

Dashboard accessible uniquement pour :

Admin



Éditeur

### 3. Protection des formulaires

Filtrage des entrées utilisateurs (htmlspecialchars, trim, etc.)

Protection contre :

injection SQL

XSS

formulaires vides

### 4. Vérification des rôles

- Un simple utilisateur **ne doit jamais accéder au panneau d'administration** sans permission.

## C. Ergonomie & Confort d'utilisation

### 1. Interface claire

Navigation simple, lisible.

Utilisation d'une police moderne (Inter).

Boutons visibles et colorés.

### 2. Responsive Design

Le site doit fonctionner sur l'ensemble des plateformes sans problème :

PC

Tablette

Smartphone

### 3. Cohérence visuelle

Couleurs principales du Fablab (bleu, turquoise, noir moderne).

Même design partout :

Projets → Articles → WebTV → Dashboard → Pages Admin.

### 4. Accessibilité

Texte lisible (contraste suffisant).

Icônes explicites.

Taille de texte adaptée.

## D. Fiabilité et Maintenabilité

### 1. Code propre et structuré

Fichiers organisés dans /app et /public.

Contrôleurs séparés par module, ex :

Pour controleur app

- AdminProjetsControleur
- AdminArticlesControleur
- etc

Pour modèles app

- AdminProjetsmodele
- AdminArticlesmodele
- etc

Pour vue app

Nom de la page demandé ex

App/Admin

- Article-admin
- login-admin
- etc

App/Parties

- header
- footer
- etc

### 2. Commentaires dans le code

Chaque fonction doit être expliquée en une courte phrase.

### 3. Base de données évolutive

Possibilité d'ajouter facilement de nouvelles colonnes.

Structure simple, lisible pour un autre développeur.

## E. Compatibilité technique

### Technologies utilisées :

**Backend** : PHP 8

**Frontend** : HTML5, CSS3, JS

**Base de données** : MySQL (PHPMYADMIN)

**Serveur local** : WAMPSEVER

### **Autres** :

- Bootstrap
  - CDN FontAwesome et Google Fonts
- 

Les besoins non-fonctionnels correspondent principalement à la qualité du site et aux exigences techniques qui lui sont liées. Le site doit donc être performant, sécurisé, facilement maintenable, mobile compatible, visuellement cohérent et conforme aux bonnes pratiques de développement web modernes.

# 5. Analyse en Pieuvre du besoin du client / Site Fablab Robotique

## 1. Description du besoin

Le projet **Fablab Robotique** répond à la nécessité de disposer d'un **site web moderne, organisé et facilement administrable**, permettant :

- de **présenter les projets** réalisés au FabLab (robotique, drones, impression 3D, électronique...),
- de **partager des articles** pédagogiques ou informatifs,
- de **diffuser des vidéos** via une WebTV intégrée,
- de proposer aux administrateurs un **espace de gestion complet** (dashboard).

Le client souhaite un outil **centralisé**, utilisable autant par les étudiants que par les formateurs, afin de :

- valoriser les travaux du FabLab,
- diffuser du contenu pédagogique,
- simplifier la communication interne/externe,
- unifier tous les modules (projets, articles, WebTV, administration) dans un même espace cohérent.

Le système doit être **simple à utiliser, sécurisé, rapide**, et permettre une mise à jour autonome du contenu, sans compétences techniques avancées.

En résumé, le besoin est de mettre en place une plateforme web **ergonomique, personnalisable et évolutive**, capable de faciliter la gestion du contenu et d'améliorer la visibilité du FabLab.

## A. Fonctions principales (FP)

La **fonction principale** (FP) est **le but général** du site.

C'est **la raison d'existence** du projet.

Le site doit permettre à différents types d'utilisateurs (visiteurs, membres, admins)

d'accéder aux informations du Fablab

de consulter des projets, articles, contenus WebTV

d'interagir avec ces contenus

et pour les admins, de gérer l'ensemble du site (backend/dashboard)

## B. Fonctions de Service (FS)

Les FS représentent toutes les fonctionnalités utiles que le site doit fournir.

### FS1 . Profil utilisateur

Le site doit permettre aux utilisateurs de créer, consulter et modifier leur profil (photo, infos personnelles, email...).

### FS2 . Gestion des articles

Le site doit permettre d'ajouter, modifier, supprimer et afficher des articles (actualité, tutoriels, news du Fablab...).

### FS3 — Gestion des projets

Le site doit afficher et permettre la création/édition/suppression de projets réalisés par les membres.

### FS4 . Web TV

Le site doit proposer un espace vidéo (streaming ou VOD) pour diffuser du contenu multimédia.

### FS5 . Upload d'images & fichiers

Les utilisateurs (selon leur rôle) doivent pouvoir envoyer des images pour projets, articles, profils.

### FS6 . Dashboard / administration

Partie backend réservée aux admins :

- gestion des utilisateurs
- gestion des articles
- gestion des projets
- gestion des images
- gestion de la sécurité

### FS7. Interface publique

Le site doit présenter un front-end moderne :

- accueil
- navigation claire

- pages projets / articles visibles par tous
- responsive
- esthétique

## C — Fonctions de Contrainte (FC)

Ce sont les obligations techniques : ce que le site **doit respecter** impérativement.

### FC1 . Sécurité / Rôles / Permissions

- protections des pages admin
- restrictions selon le rôle (visiteur, membre, admin)
- protection contre intrusions, injections SQL, etc.

### FC2 . Base de données (intégrité et cohérence)

- base de données structurée
- données cohérentes
- respect des dépendances
- sauvegarde correcte

### FC3 . Compatibilité appareils

- site adapté PC / tablette / mobile
- responsive design

### FC4 . Performance

- temps de chargement rapides
- gestion correcte des images
- optimisation des requêtes SQL

### FC5 . Upload sécurisé

- contrôle du type de fichiers
- limite de taille
- blocage fichiers dangereux

### FC6 . Sessions / Authentification sécurisées

- mots de passe hashés
- sessions protégées
- prévention des usurpations d'identité

### FC7 . Maintenance & évolutivité

Le site doit être facile à maintenir :

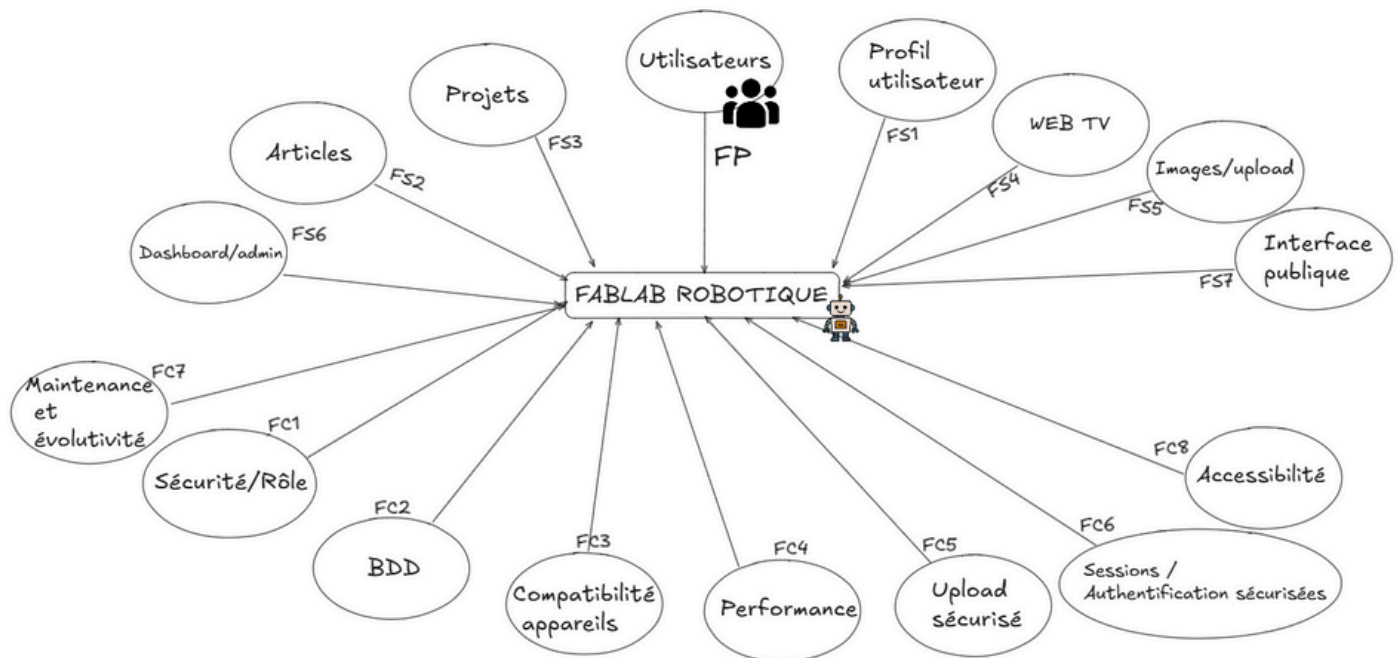
- code clair

- architecture MVC
- fichiers organisés
- réutilisabilité du code

## FC8 — Accessibilité

- compatibilité clavier
- contraste lisible
- navigation simple pour tous les utilisateurs

### DIAGRAMME PIEUVRE





# 6. Analyse Fonctionnelle plus clair (FP-FS,FC)

Cette analyse décrit comment le site Fablab Robotique répond aux besoins exprimés.

On décompose les fonctions principales en sous-fonctions simples, organisées de manière logique.

## Fonction Principale

Permettre aux utilisateurs d'accéder, consulter et gérer les contenus du Fablab Robotique. Cette fonction principale est remplie grâce à plusieurs sous-fonctions décrites ci-dessous.

## Fonctions de Service

### FS1. Gestion des utilisateurs

- FS1.1 : Création d'un compte utilisateur
- FS1.2 : Connexion / Déconnexion
- FS1.3 : Gestion du profil (nom, email, image)
- FS1.4 : Définition et gestion des rôles (admin, éditeur, membre)

### FS2. Gestion des articles

- FS2.1 : Création d'un article (titre, contenu, image)
- FS2.2 : Modification d'un article
- FS2.3 : Suppression d'un article
- FS2.4 : Affichage public des articles
- FS2.5 : Upload d'images pour articles

### FS3. Gestion des projets

- FS3.1 : Création de projets (titre, description, technologies, image)
- FS3.2 : Modification / suppression des projets
- FS3.3 : Affichage détaillé d'un projet
- FS3.4 : Catégorisation automatique (robotique, drone...)
- FS3.5 : Upload d'images de projets

### FS4. Module Web TV

- FS4.1 : Affichage des vidéos (Web TV)
- FS4.2 : Gestion des vidéos en Back-office
- FS4.3 : Intégration d'iframe ou de lien externe Youtube/Drive

## **FS5. Upload / Gestion des fichiers**

- FS5.1 : Upload sécurisé (projets, articles, profils)
- FS5.2 : Format accepté (PNG, JPG, WEBP)
- FS5.3 : Stockage des fichiers dans le serveur
- FS5.4 : Vérification automatique (type MIME, taille, extension)

## **FS6. Dashboard / Administration**

- FS6.1 : Panneau administrateur
- FS6.2 : Gestion des utilisateurs
- FS6.3 : Gestion des articles
- FS6.4 : Gestion des projets
- FS6.5 : Gestion des images / uploads
- FS6.6 : Accès sécurisé selon les permissions

## **FS7. Interface publique**

- FS7.1 : Page d'accueil
- FS7.2 : Navigation claire (menu, header, footer)
- FS7.3 : Pages articles
- FS7.4 : Pages projets
- FS7.5 : Interface responsive
- FS7.6 : Esthétique moderne (couleurs, typographie, animations)

# **Fonctions de Contrainte**

## **FC1. Sécurité**

- FC1.1 : Hashage des mots de passe
- FC1.2 : Vérification des permissions
- FC1.3 : Protection contre les injections SQL
- FC1.4 : Gestion sécurisée des sessions

## **FC2. Base de données**

- FC2.1 : Structure cohérente (tables utilisateurs, projets, articles...)
- FC2.2 : Intégrité référentielle
- FC2.3 : Sauvegarde correcte des données
- FC2.4 : Modèle MVC respecté

## **FC3. Compatibilité des appareils**

- FC3.1 : Site responsive (mobile, tablette, PC)
- FC3.2 : Composants adaptatifs
- FC3.3 : Media queries optimisées

## **FC4. Performance**

- FC4.1 : Optimisation des requêtes SQL
- FC4.2 : Mise en cache simple (images / assets)
- FC4.3 : Compression des fichiers statiques
- FC4.4 : Chargement rapide

## **FC5. Upload sécurisé**

- FC5.1 : Contrôle des extensions
- FC5.2 : Limitation de taille
- FC5.3 : Suppression fichiers dangereux
- FC5.4 : Stockage isolé des images utilisateur

## **FC6. Sessions sécurisées**

- FC6.1 : Gestion des cookies
- FC6.2 : Prévention des usurpations de session
- FC6.3 : Expiration contrôlée des sessions

## **FC7. Maintenance & évolutivité**

- FC7.1 : Structure MVC
- FC7.2 : Code organisé et commenté
- FC7.3 : Ajout facile de nouvelles sections
- FC7.4 : Fichiers modulaires

## **FC8. Accessibilité**

- FC8.1 : Navigation clavier
- FC8.2 : Contraste lisible
- FC8.3 : Structure sémantique correcte (balises HTML)
- FC8.4 : Zones interactives suffisamment grandes

# 7. Le cas d'Utilisateurs

## Acteur 1 : Visiteur (non connecté)

### Consulter les projets

En tant que *visiteur*

*Je veux voir la liste des projets affichés sur le site, Afin de découvrir les réalisations du FabLab.*

### Voir le détail d'un projet

*Je veux ouvrir la fiche complète d'un projet, Afin de connaître sa description, son image et les technologies utilisées.*

### Consulter les articles

*Je veux voir les articles publiés, Afin de lire des informations et des actualités.*

### Voir une vidéo WebTV

*Je veux visionner une vidéo présente dans la section WebTV, Afin de accéder aux contenus multimédias du FabLab.*

## Acteur 2 : Membre (Utilisateur connecté)

### Modifier mon profil

En tant que *membre*

*Je veux modifier mes informations personnelles (email, nom, photo), Afin de garder un profil à jour.*

### Upload d'avatar / image

*Je veux importer une image (avatar), Afin de personnaliser mon profil.*

### Accéder aux pages réservées

*Je veux accéder aux sections disponibles selon mon rôle, Afin d'utiliser des fonctions avancées du site.*

## Acteur 3 : Éditeur

*Gère le contenu (Article et projet) mais pas les utilisateurs*

### **Créer un article**

*En tant qu'éditeur*

*Je veux rédiger un article avec une image, Afin de publier du contenu pour les visiteurs.*

### **Modifier un article**

*Je veux mettre à jour le contenu d'un article, Afin de corriger ou améliorer les informations.*

### **Créer un projet**

*Je veux ajouter un nouveau projet, Afin de partager mes créations.*

### **Modifier / supprimer un projet**

*Je veux mettre à jour ou retirer un projet, Afin d'assurer une cohérence du contenu.*

## **Acteur 4 : Administrateur (admin du site)**

### **Accéder au Dashboard**

*En tant qu'admin, Je veux accéder à un tableau de bord complet, Afin de gérer l'ensemble des modules du site.*

### **Gérer les utilisateurs**

*Je veux voir la liste des utilisateurs et leurs rôles, Afin de contrôler qui peut faire quoi.*

### **Supprimer un utilisateur**

*Je veux retirer un compte du système, Afin de garantir la sécurité et la propreté de la base.*

### **Gérer les projets**

*Je veux voir, modifier, supprimer ou ajouter des projets, Afin de garder une base cohérente.*

### **Gérer les articles**

*Je veux contrôler tous les articles du site, Afin de valider ou corriger le travail des éditeurs.*

### **Gérer les vidéos WebTV**

*Je veux ajouter, modifier ou supprimer une vidéo, Afin de piloter la partie multimédia.*

## Gérer les images / fichiers uploadés

Je veux *pouvoir supprimer des images inutiles*, Afin de *maîtriser l'espace du serveur*.

# 8. Description détaillée des pages du Site Fablab Robotique (en cours)

## Objectifs

Présenter le FabLab, ses activités, ses projets récents et guider l'utilisateur vers les sections importantes.

## 8.1 Page d'Accueil (Home)

### Objectifs

Présenter le FabLab, ses activités, ses projets récents et guider l'utilisateur vers les sections importantes.

### Contenu principal

- Accès rapide :  
*Projets – Articles – WebTV – Contact – Login.*
- *résumé Fablabrobotique*
- Bandeau d'accueil avec titre .
- Aperçu des derniers projets.
- Aperçu des derniers articles.
- Effets visuels (particules, animations légères au niveau des cards ).

### Visiteurs concernés

Tous les utilisateurs (visiteurs, membres, admin).

## 8.2 Page Projets (liste)

### Objectifs

Afficher l'ensemble des projets réalisés au FabLab avec possibilité de filtre/recherche.

### Fonctionnalités

- Grille de cartes projets (image + titre + résumé).
- Système de catégories automatiques  
(robotique, drone, électronique, autres...).

- Barre de recherche instantanée (JS).
- Filtre par catégorie.
- Bouton "Créer un projet" (réservé aux Admin/Éditeurs).
- Image affichée depuis :
  - un upload (serveur)
  - ou une URL externe

## Rôles

- Visiteur : consultation uniquement
- Éditeur/Admin : peut ajouter, modifier, supprimer.

## 8.3 Page Détail d'un Projet

### Objectifs

Présenter un projet complet.

### Contenu

- Grande image du projet  
(*plein rectangle comme tu veux l'afficher*)
- Titre + résumé + technologies utilisées
- Description détaillée
- Tags
- Date de création / dernière mise à jour
- Bouton *Voir la galerie* (si nécessaire)
- Bouton retour
- Mode modal (fenêtre) depuis la liste

### Accès

Tous les utilisateurs peuvent voir.

## 8.4 Page Articles (liste)

### Objectifs

Afficher les articles pédagogiques, tutoriels, news...

### Fonctionnalités

- Liste en cartes avec image (upload ou URL)
- Titre + extrait + auteur + date
- Barre de recherche (JS)
- Filtre (catégorie si ajouté plus tard)
- Bouton "Créer un article" (Admin/Éditeur)



- Chargement cohérent des images comme dans la page projets  
(même système d'affichage : ../public/images/...)

## Rôles

- Visiteur : lecture
- Éditeur/Admin : gestion complète

# 8.5 Page Détail d'un Article

## Objectifs

Afficher le contenu complet d'un article.

## Contenu

- Grande image (même style que page projet)
- Titre + auteur + date + tags
- Corps de l'article (HTML avec sauts de ligne)
- Zone "informations complémentaires" :

-

- résumé
- tags
- catégorie

-

- Bouton retour
- Sidebar d'information
- Icône alternative si image absente ou invalide

# 8.6 Page WebTV

## Objectifs

Diffuser des vidéos liées au FabLab.

## Contenu

- Lecteur vidéo intégré (embed YouTube/Vimeo ou fichier)
- Liste des dernières vidéos
- Filtres simple (type de vidéo)
- Bouton "Ajouter une vidéo" (si admin)
- Description de chaque vidéo

## Rôles

- Visiteur : visionnage
- Admin : gestion des vidéos

## 8.7 Page Profil utilisateur

### Objectifs

Centraliser les informations de l'utilisateur connecté.

### Fonctionnalités

- Informations personnelles : nom, prénom, email
- Photo de profil (upload)
- Rôle (Visiteur, Éditeur, Admin)
- Historique (derniers contenus)
- Paramètres (mot de passe, préférences)
- Bouton déconnexion

## 8.8 Page Contact

en cours

## 8.9 Page de Connexion / Inscription

### Objectifs

Permettre à l'utilisateur d'accéder à son espace.

### Fonctionnalités

- Formulaire de connexion
- Formulaire d'inscription

Validation :

- Email réel
- Mot de passe sécurisé (longueur, hashage)
- Gestion erreurs
- Redirection après connexion

## 8.10 Dashboard Administrateur

## **Objectifs**

**Gérer tout le contenu interne du site.**

## **Modules**

1. **Gestion des utilisateurs**
  - **ajouter / modifier / supprimer**
  - **changer rôle (visiteur / éditeur / admin)**
2. **Gestion des projets**
  - **CRUD**
  - **Upload image**
  - **Aperçu**
  - **Recherche interne**
3. **Gestion des articles**
  - **CRUD**
  - **Upload image possible OU URL externe**
  - **Éditeur de texte simple**
4. **Gestion WebTV**
  - **ajout vidéos**
  - **modification**
  - **suppression**
5. **Statistiques générales**
  - **nombre projets**
  - **nombre articles**
  - **activité récente**
6. **Sécurité**
  - **restrictions par rôle**
  - **pages protégées**
  - **déconnexion**

## **Accès**

**Uniquement admin (et parfois éditeur).**



