



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche

Université des Sciences et de la Technologie Houari Boumediene



# Rapport Projet 6 : Génération automatique des formulaires

## Préparé par:

Mouaci Rabie

Khenouf hamza

Belakab Billal

Dahmouche Melissa Ahlem

Akkouchi Nesrine

Tayebi Souad

## Encadrant:

Mm Guendouz

Année universitaire: 2024-2025



# Chapitre 1 : Introduction de Notre Solution

## 1. Problématique :

À l'Université des Sciences et de la Technologie Houari-Boumediene (USTHB), les documents administratifs et pédagogiques (attestations, demandes, formulaires de stage, etc.) sont publiés via la **plateforme ENT**, le système d'information de l'université. Cependant, leur création reste un processus manuel, fastidieux et sujet à des erreurs.

Actuellement, ces formulaires sont conçus à partir de modèles Word ou PDF peu flexibles, nécessitant une saisie répétitive et des ajustements constants. Bien que SIT permette leur diffusion, **l'absence d'une solution automatisée** et intégrée pour leur génération entraîne plusieurs problèmes :

- **Temps perdu** : La conception et la mise en page manuelles ralentissent la production des documents.
- **Risques d'erreurs** : La saisie répétitive augmente les incohérences et les fautes.
- **Manque de standardisation** : Les formulaires varient en structure et en design, nuisant à la cohérence institutionnelle.
- **Difficulté de gestion** : La réutilisation et la modification des modèles existants ne sont pas optimisées.

Pour améliorer ce processus, l'intégration d'un **module intelligent de génération de formulaires** directement dans SIT permettrait d'automatiser la création, d'uniformiser les formats et de réduire les tâches manuelles. Une telle modernisation renforcerait l'efficacité administrative tout en alignant l'USTHB sur les meilleures pratiques de digitalisation.

## 2. Solution Proposée :

Notre solution, **USTHB FORMS**, vise à résoudre ces problèmes en proposant une plateforme intelligente de génération automatique de formulaires.



USTHB FORMS est une application web qui permet:

- Aux “form owner” de créer facilement des formulaires standardisés
- Une génération assistée par IA pour suggérer des champs pertinents
- Une gestion centralisée des brouillons (drafts) et formulaires publiés(published)
- Une interface unifiée pour tous les utilisateurs (administrateurs, propriétaires de formulaires et répondants)

### **3. Structure des Utilisateurs :**

#### **A. Propriétaires de formulaires (Form Owners):**

- a. Connexion sécurisée
- b. Création de formulaires (manuellement ou via IA)
- c. Gestion des brouillons et publications
- d. Accès aux statistiques des réponses

#### **B. Administrateurs :**

- a. Visualisation de tous les formulaires
- b. Accès aux statistiques des réponses de tous Propriétaires de formulaires
- c.
- d.

#### **C. Répondants (Form Fillers):**

- a. Interface simplifiée pour compléter les formulaires
- b. Expérience utilisateur optimisée

# Chapitre 2 : Conception.

## 1. Analyse des Besoins:

Avant de commencer la conception, une analyse approfondie des besoins fonctionnels et non-fonctionnels est nécessaire. Voici les principaux besoins identifiés :

- **Besoins fonctionnels :**

- Génération automatique des formulaires via AI et les autres outils.
- Consultation des réponses par les Propriétaires de formulaires et les administrateurs.
- Stockages des formulaires ( brouillons ou publiés )
- Génération des liens pour les formulaires publiés

- **Besoins non-fonctionnels :**

- Performance et rapidité de génération.
- Sécurité des données
- Interface utilisateur conviviale.

## 2. Choix des Technologies :

### 3. Choix des Technologies de Frontend :

- Framework** : Next.js (pour le rendu côté serveur et une expérience performante)
- Langage** : TypeScript (pour un code plus robuste et typé)
- Stylage** : Tailwind CSS (pour un design rapide et responsive)
- Design & Prototypage** : Figma (pour les maquettes UI/UX et les tests utilisateurs)

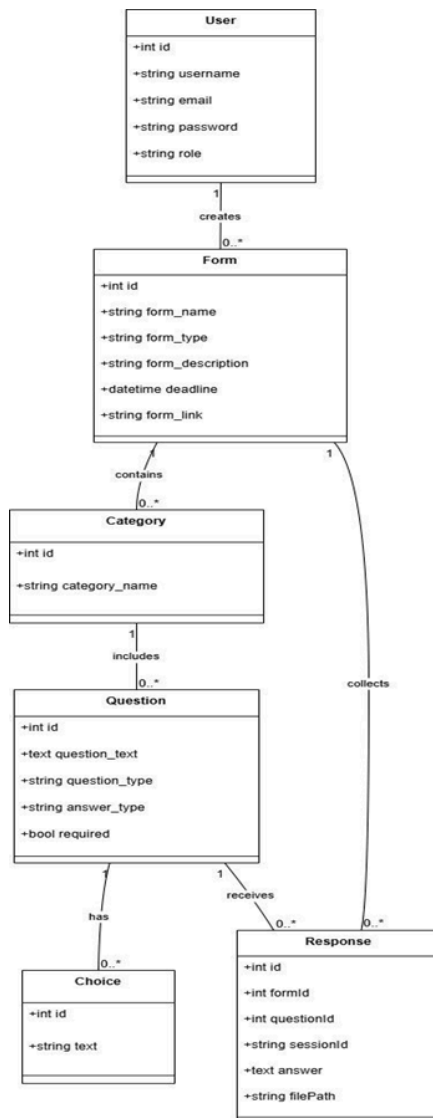
#### 4. Choix des Technologies pour le Backend & Base de Données:

La gestion des données structurées est assurée par **‘MySQL’**, choisi pour ses qualités éprouvées en production :

## Pour la Base de Données:

- a. **Intégrité référentielle** : contraintes de clés étrangères (ON DELETE CASCADE, ON UPDATE RESTRICT).
- b.
- c. **Transactions ACID** : garantie de cohérence et d'atomicité des opérations critiques.
- d.
- e. **Transactions ACID** : garantie de cohérence et d'atomicité des opérations critiques.
- f.
- g. **Optimisation**: indexation adaptée (user\_id, form\_id, question\_id) et analyse EXPLAIN pour ajustement des requêtes.
- h.
- a. **Haute disponibilité** : sauvegardes automatisées et réplication maître-esclave.
- b. L'ORM 'Sequelize' facilite :
  - Définition des modèles et associations (hasMany, belongsTo) de manière déclarative.
  - Génération et exécution de migrations versionnées.
  - Construction de requêtes complexes (eager/lazy loading, agrégations) avec une API fluide.

### Schéma UML de la base de données:



## Pour le Backend:

Le backend, structuré en **JavaScript ES6** avec **Node.js** et **Express**, repose sur une architecture en couches:

Routes → Contrôleurs → Services → Modèles



Ce découpage assure :

- d. **Lisibilité et séparation** des préoccupations.
- e. **Lisibilité et séparation** des préoccupations.
- f. **Testabilité** : services testés isolément via Mocha/Chai.
- g. **Scalabilité** : ajout de fonctionnalités sans impact sur la structure existante.

## Technologies et librairies :

- i. **‘Multer’** pour les uploads multipart/form-data, stockés sur **\*\*Cloudflare R2\*\***.
- j. **‘Swagger UI’** pour la documentation interactive de l’API.
- k. **‘Joi’** pour la validation stricte des payloads.
- l. **‘Winston’** et **‘Logrotate’** pour la gestion avancée des logs.

## 7. Choix des Technologies pour le AI :

Après une étude approfondie des besoins des utilisateurs finaux, nous avons identifié les exigences :

- a. **Flask** : Framework web Python léger et flexible qui nous permet de créer rapidement des API RESTful. Flask est utilisé pour gérer toutes les routes, les requêtes et les réponses de notre application.
- b.
- c. **Google Generative AI (Gemini)** : Modèle d'IA avancé de Google (Gemini 2.0 Flash) qui alimente nos fonctionnalités de génération de

formulaire et de conversation. Ce modèle est particulièrement efficace pour comprendre le contexte et générer du contenu structuré.

- d.
- e. **RegEx (Expressions Régulières)** : Utilisé pour extraire et analyser proprement les réponses JSON provenant du modèle d'IA, même lorsqu'elles sont encapsulées dans des blocs de code.

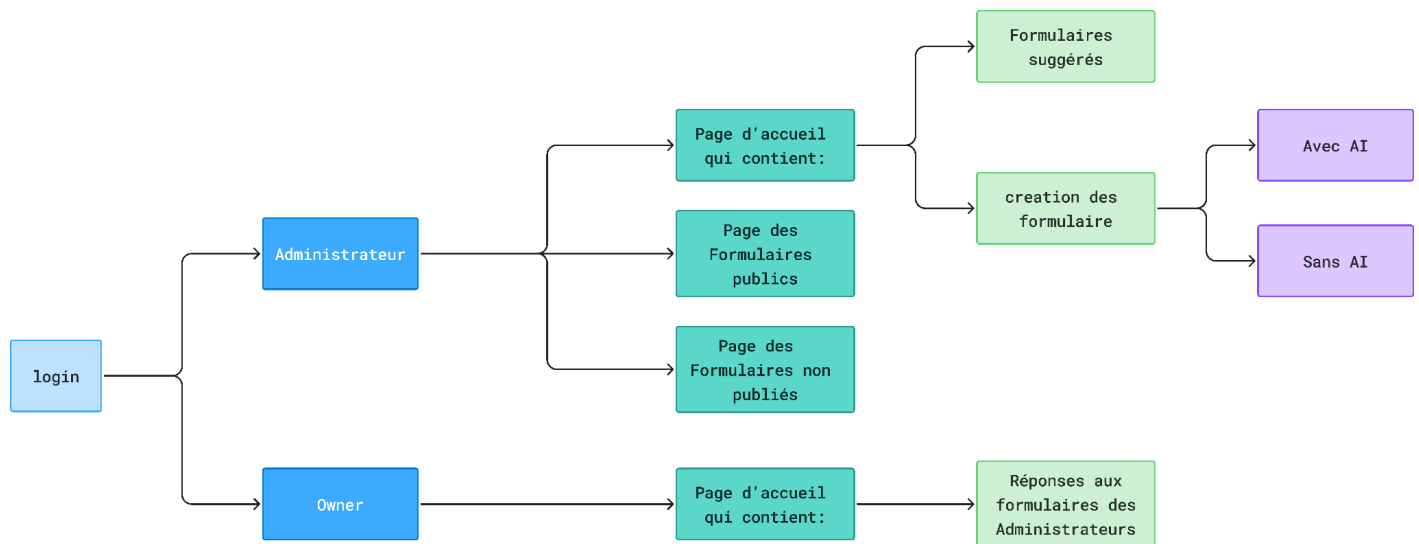
8.

- a. **Gestion d'État Conversationnel** : Système personnalisé pour maintenir l'historique des conversations et l'état des formulaires entre les requêtes, permettant une expérience interactive et itérative.

b.

c.

- **Architecture du Système:**



## Interactions des Composants:

### 1. Fonctionnalités Frontales :

#### ● Flux Principal d'Authentification:

- Écran de connexion unique avec sélection du profil (Administrateur/Propriétaire)
- Redirection vers l'interface appropriée selon le rôle

#### ● Espace des utilisateurs :

##### 1. Espace Propriétaire:

##### ❖ Page "Créer un formulaire" (3 modes de création):

##### Mode Formulaires Suggestés :

- ❖ Galerie de modèles pré-configurés (inscriptions, évaluations, enquêtes...)
- ❖ Prévisualisation interactive avant sélection

- ❖ Interface de chat conversationnel intégrée:
  - Personnalisation des champs existants
  - Ajout de nouvelles sections/questions
  - Options de publication (immédiate/différée)

### **Mode Assisté par IA:**

- ❖ Interface de chat conversationnel intégrée:
  - L'administrateur décrit ses besoins en langage naturel
  - L'IA pose des questions complémentaires pour affiner sa compréhension
  - Génération progressive de la structure du formulaire
- ❖ Éditeur intelligent avec:
  - Suggestions contextuelles pendant la modification
  - Validation automatique de la cohérence
  - Bouton "Appliquer les modifications" pour intégrer les changements

### **Mode Manuel Avancé:**

- ❖ Constructeur visuel avec::
  - Bibliothèque de sections types (informations personnelles, choix multiples...)
  - Palette d'éléments formulaires (champs texte, cases à cocher, menus déroulants)
  - Système de glisser-déposer pour l'agencement
- ❖ Fonctionnalités d'édition:
  - Ajout/suppression de sections
  - Configuration fine des questions (obligatoire, type de réponse...)
  - Prévisualisation en temps réel

### **❖ Page " Gestion des Publications" :**

- ❖ Tableau de bord des formulaires publiés:
  - Statut (actif/expiré)
  - Métriques (nombre de réponses, taux de complétion)
  - Actions rapides (désactiver, dupliquer, archiver)

- ❖ Accès aux réponses:
  - Interface de configuration des visualisations
  - Filtres intelligents par critères
  - Export des données (Sheet)

### ❖ Page "Mes Drafts":

- ❖ Système de versioning des brouillons
- ❖ Fonctions d'édition:
  - Reprise du travail où il a été laissé
  - Comparaison avec version précédente
  - Suppression/duplication

## 2. Espace Administrateur:

### 3. Espace Répondants (Form Fillers):

- Accès au formulaire avec le lien de formulaire
- Répondre aux questions
- Envoyer la réponse

- ❖ Bouton "Voir les réponses" générant:
  - Vue synthétique (graphiques, statistiques)
  - Accès au tableau complet des données
  - Outils d'analyse intégrés

- **Fonctionnement de l'API:**

- 1. Espace Validation du Contexte :**

- ❖ **Page Chaque requête utilisateur est d'abord validée pour s'assurer qu'elle concerne bien des formulaires universitaires. Cette validation est intentionnellement souple pour permettre une grande variété de cas d'usage académiques tout en filtrant les contenus problématiques.**

- 2. Conversation Intelligente :**

- ❖ L'API maintient une conversation contextuelle avec l'utilisateur pour comprendre précisément ses besoins:
  - ❖ Les requêtes initiales sont analysées pour déterminer le type de formulaire souhaité
  - ❖ Des questions de suivi pertinentes sont générées pour recueillir plus de détails
  - ❖ L'historique complet de la conversation est pris en compte pour chaque nouvelle interaction

- 3. Préservation de Structure:**

- ❖ Un aspect crucial de notre API est sa capacité à préserver la structure existante des formulaires:
  - ❖ L'état actuel du formulaire est transmis avec chaque requête

- ❖ Le modèle d'IA reçoit des instructions explicites pour conserver les catégories et questions existantes
- ❖ Un système de vérification garantit que les éléments existants ne sont jamais perdus ou modifiés

#### **4. Génération de Formulaires:**

- ❖ La génération finale produit une structure JSON complète et bien organisée:
- ❖ Les formulaires sont divisés en catégories logiques
- ❖ Chaque question inclut des métadonnées appropriées (type, obligatoire/optionnel, etc.)
- ❖ Les formats de réponse sont spécifiés (texte, nombre, date, email, booléen)
- ❖ Les questions à choix multiple incluent des options prédéfinies pertinentes

#### **● Points de Terminaison (Endpoints) Principaux :**

Notre API expose trois endpoints essentiels:

##### **1. /chat (POST):**

- ❖ L'endpoint /chat est le cœur de l'interaction intelligente avec l'utilisateur. Il:

- ❖ Maintient l'historique des conversations via un ID unique (conservation\_id)
- ❖ Génère des questions pertinentes pour préciser les besoins
- ❖ Peut recevoir et prendre en compte l'état actuel du formulaire
- ❖ S'adapte progressivement pour construire une compréhension complète des exigences

## 2. /generate (POST) :

- ❖ L'endpoint /generate est responsable de la création finale du formulaire. Ses caractéristiques clés:

- ❖ Il utilise tout l'historique des conversations pour contextualiser la génération
- ❖ Il préserve et intègre intelligemment toute structure de formulaire existante
- ❖ Il produit une structure JSON complète prête à être utilisée par l'interface utilisateur
- ❖ Il inclut des validations pour garantir que le formulaire généré respecte les standards requis.

## ● Description détaillée de la base de données:

### ○ Table 'users' :

- ❖ id (INT) - PK, AUTO\_INCREMENT : Identifiant unique de l'utilisateur
- ❖ username (VARCHAR(255)) - NOT NULL : Nom d'utilisateur affiché



- ❖ email (VARCHAR(255)) - UNIQUE, NOT NULL : Adresse email utilisée pour l'authentification
- ❖ password (VARCHAR(255)) - NOT NULL : Mot de passe hashé de l'utilisateur

○ **Table 'forms' :**

- ❖ id (INT) - PK, AUTO\_INCREMENT : Identifiant unique du formulaire
- ❖ - form\_name (VARCHAR(255)) - NOT NULL : Nom ou titre du formulaire
- ❖ - form\_type (VARCHAR(255)) - NOT NULL : Type de formulaire (ex: RH, Sondage)
- ❖ - form\_description (VARCHAR(255)) - NULLABLE : Description optionnelle du formulaire
- ❖ - deadline (TIMESTAMP) - NULLABLE : Date limite de soumission du formulaire
- ❖ - form\_link (VARCHAR(255)) - NULLABLE : Lien URL vers le formulaire

○ **Table 'categorie' :**

- ❖ - id (INT) - PK, AUTO\_INCREMENT : Identifiant unique de la catégorie
- ❖ - category\_name (VARCHAR(255)) - NOT NULL : Nom de la catégorie

○ **Table 'questions' :**

- ❖ id (INT) - PK, AUTO\_INCREMENT : Identifiant unique de la question
- ❖ - question\_text (TEXT) - NOT NULL : Texte de la question posée
- ❖ - question\_type (VARCHAR) - NOT NULL, DEFAULT 'text' : Type de question (texte, select, etc.)
- ❖ - answer\_type (VARCHAR(50)) - NOT NULL, DEFAULT 'text' : Format attendu de la réponse

- ❖ - required (BOOLEAN) - NOT NULL, DEFAULT false : Indique si la réponse est obligatoire

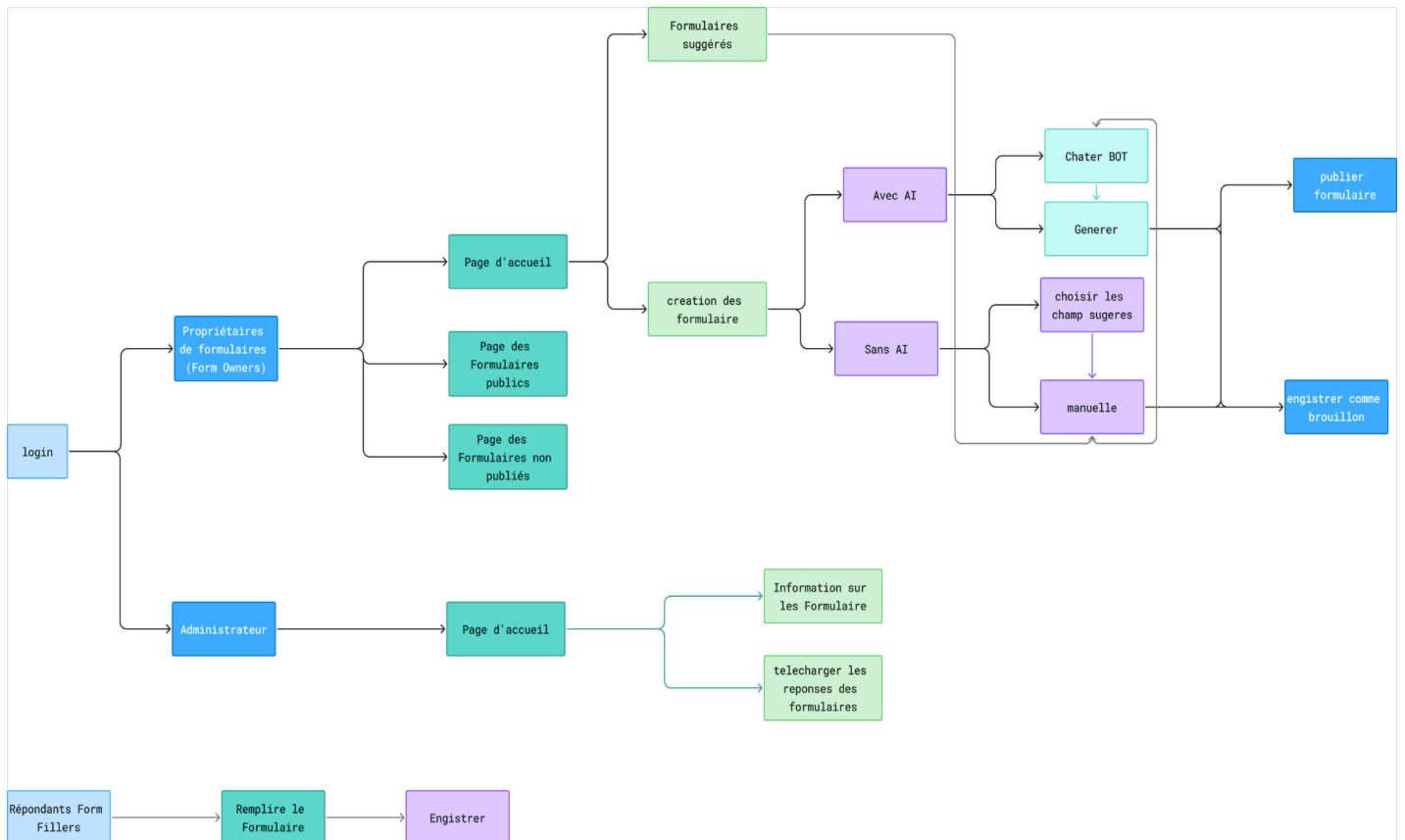
- **Table ‘choices’:**

- ❖ - id (INT) - PK, AUTO\_INCREMENT : Identifiant unique du choix
- ❖ - text (VARCHAR) - NOT NULL : Texte du choix proposé

- **Table ‘responses’ :**

- ❖ id (INT) - PK, AUTO\_INCREMENT : Identifiant unique de la réponse
- ❖ formId (INT) - NOT NULL : Référence au formulaire concerné
- ❖ questionId (INT) - NOT NULL : Référence à la question répondue
- ❖ sessionId (VARCHAR(36)) - NOT NULL : Identifiant anonyme de session utilisateur
- ❖ answer (MEDIUMTEXT) - NULLABLE : Réponse textuelle à la question
- ❖ filePath (VARCHAR(255)) - NULLABLE : Lien vers un fichier si upload
- ❖ Contraintes avancées (exemple) :
  - ➔ ```sql
  - ➔ ALTER TABLE responses
  - ➔ ADD CONSTRAINT fk\_question
  - ➔ FOREIGN KEY (questionId)
  - ➔ REFERENCES questions(id)
  - ➔ ON DELETE CASCADE;
  - ➔ ```

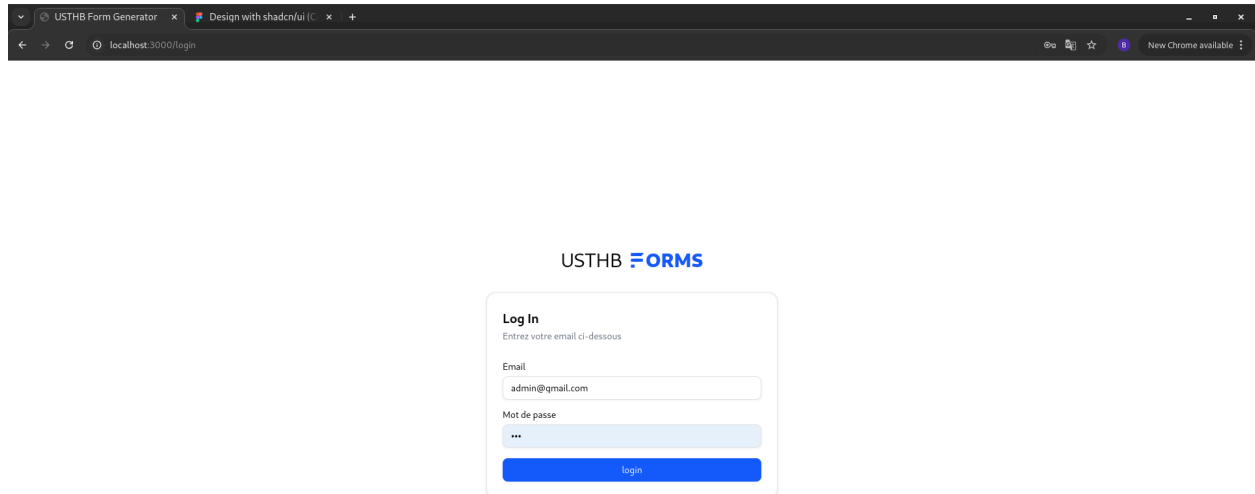
- **User Flow :**



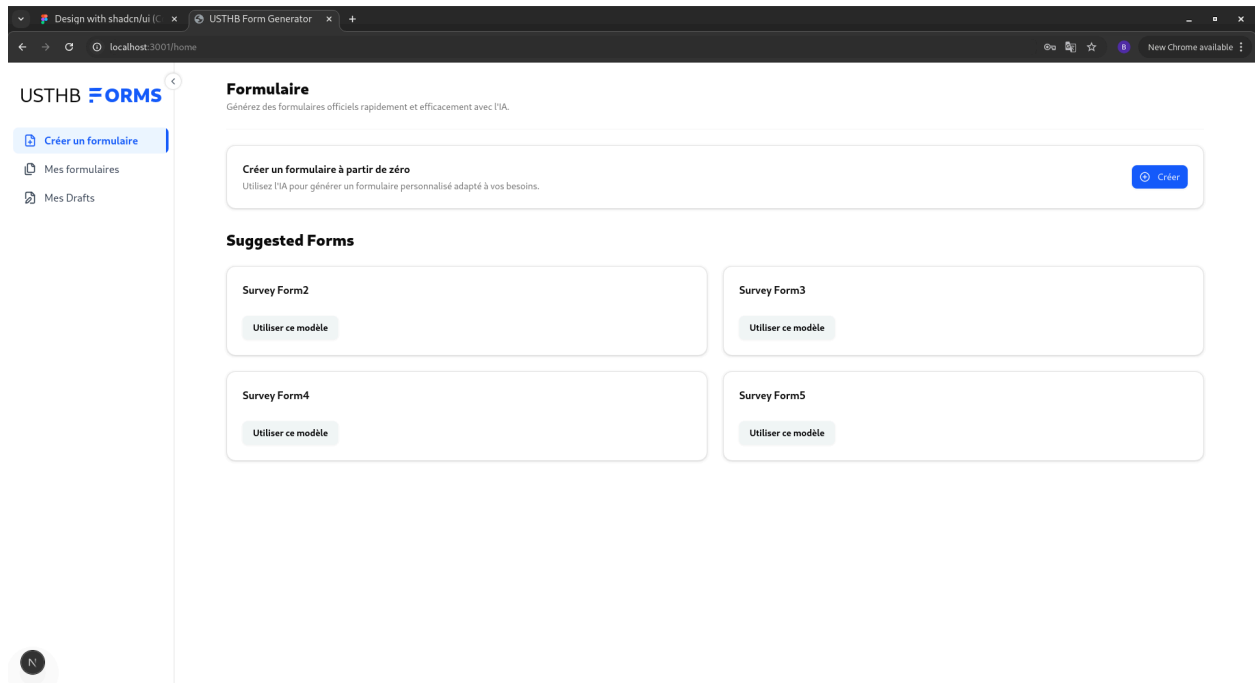
---

## Chapitre 3 : Implementation.

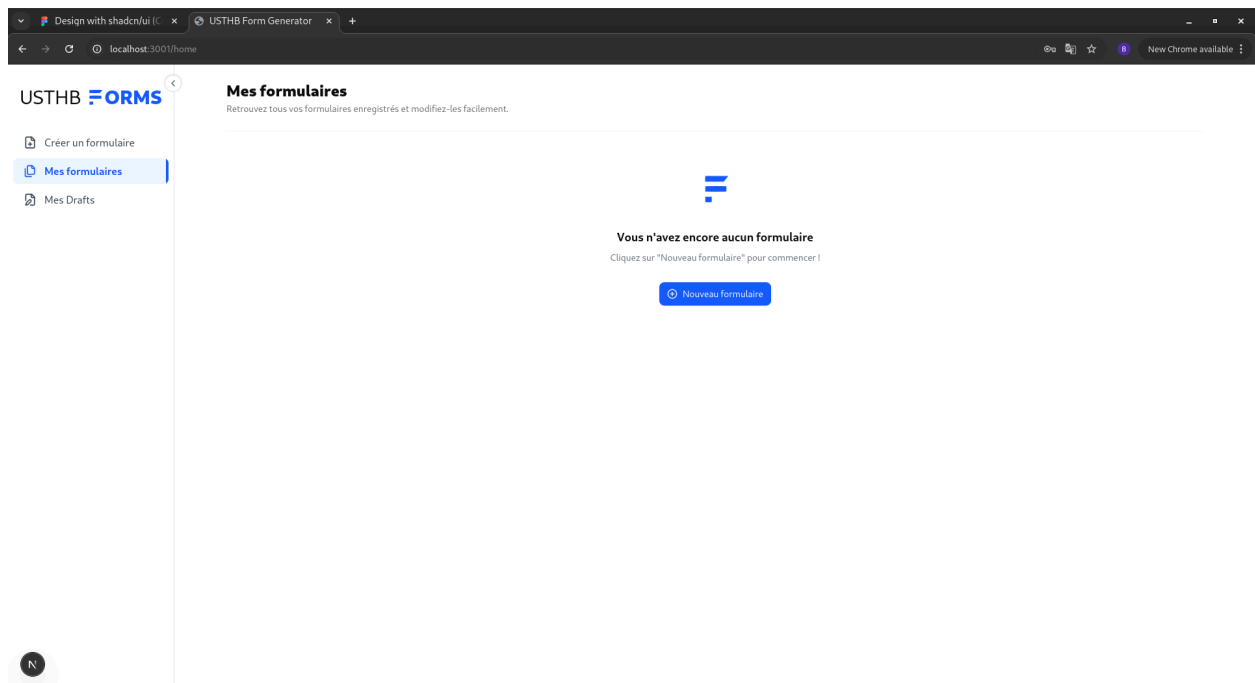
### Présentation des Écrans de notre Site Web Formulaire USTHB

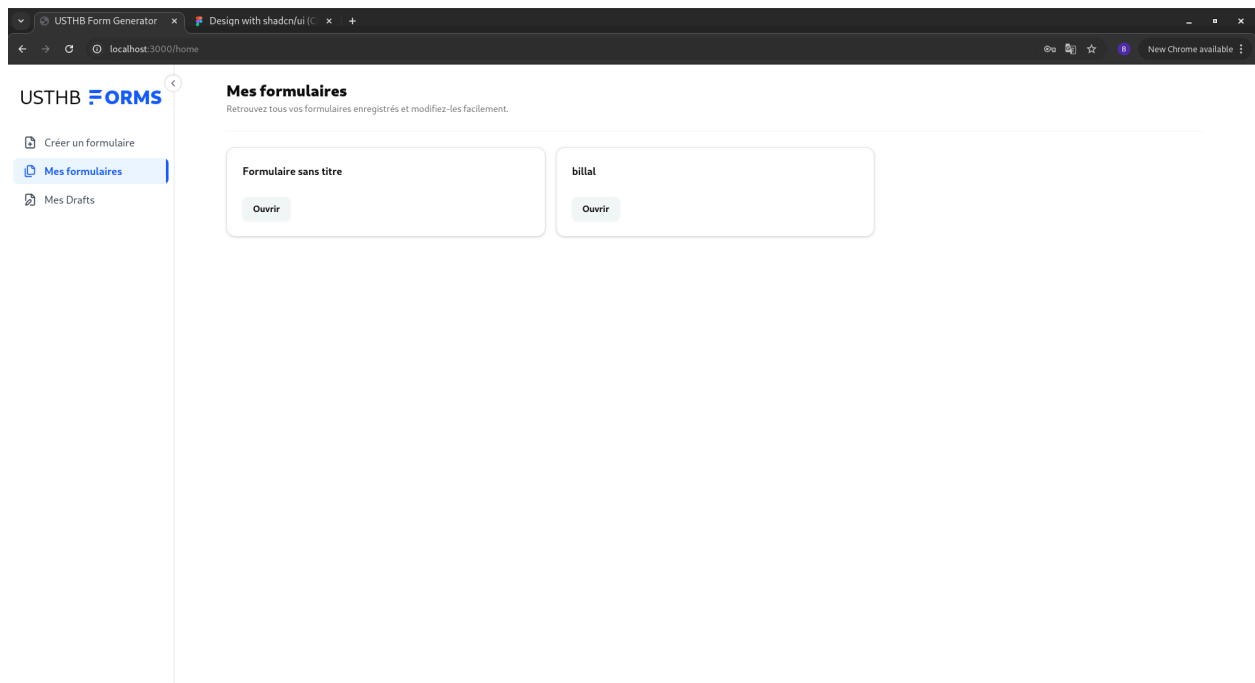


Page d'accueil Propriétaires de formulaires (Form Owners)

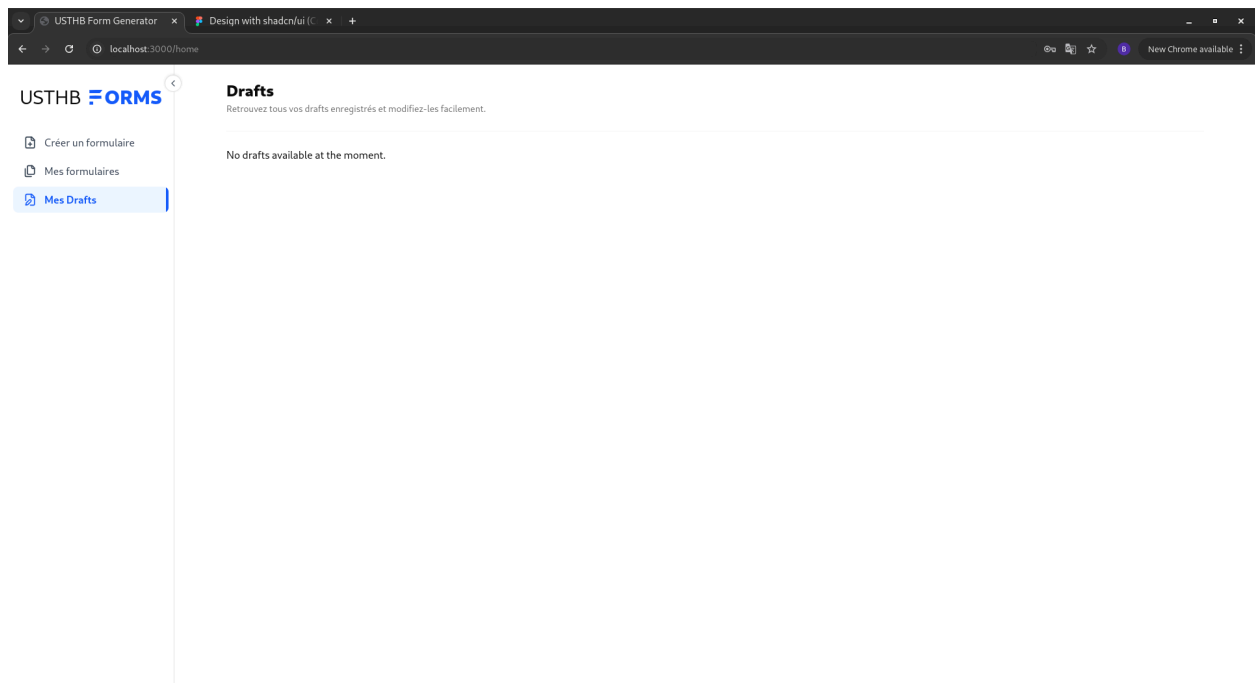


## Page des Formulaires publics

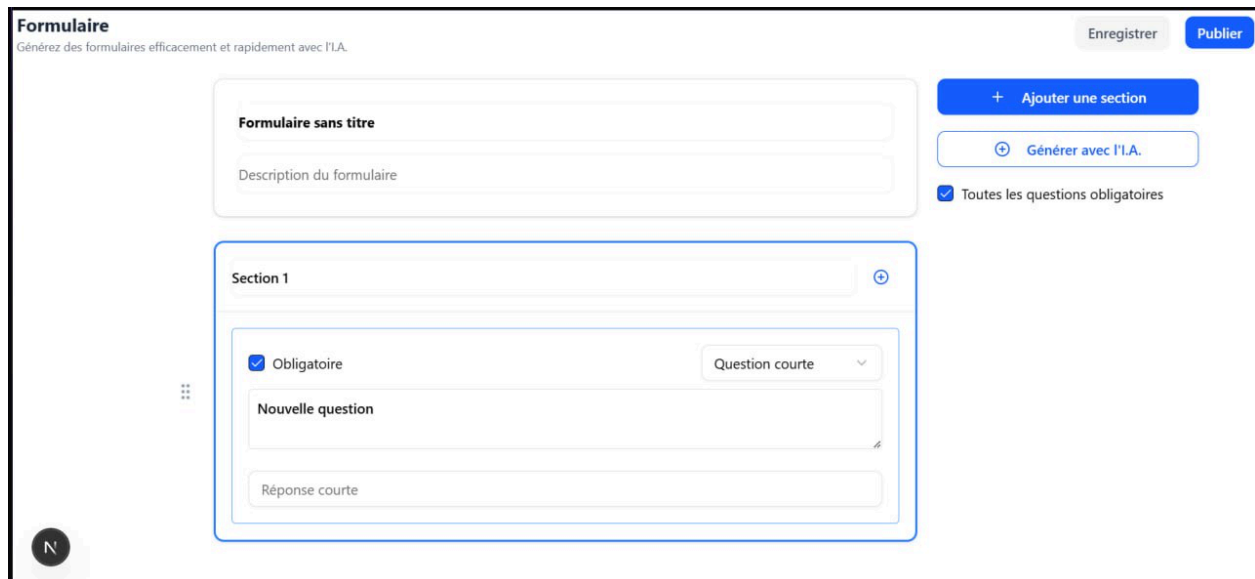
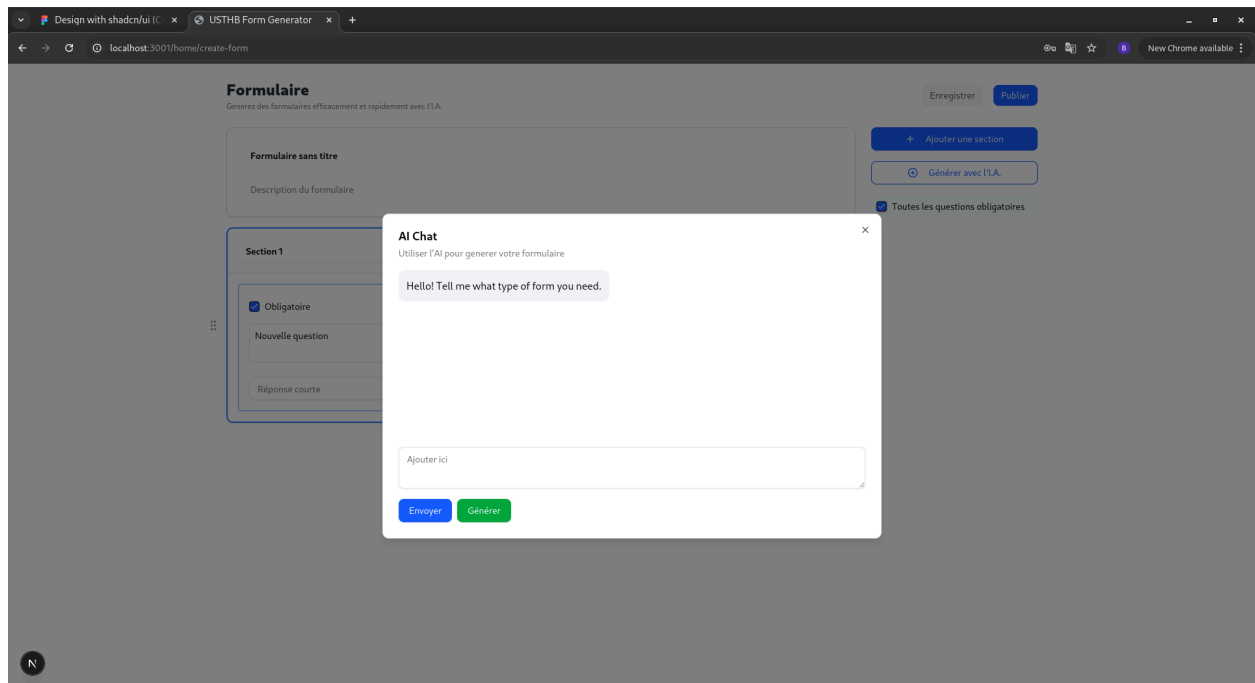




Page des Formulaires non publics



Création des formulaire avec AI



Création des formulaire sans AI



Formulaire

Construisez votre formulaire section par section selon vos besoins, sans utiliser l'IA.

USTHB FORM AI

Ajouter une section

Name

Matricule

Matricule

Générer le Formulaire

## Création des Formulaires

Formulaire

Générez des formulaires efficacement et rapidement avec l'IA.

EnregistrerPublier

Formulaire sans titre

Description du formulaire

+ Ajouter une section

Générer avec l'IA.

☒ Toutes les questions obligatoires

Section 1

☒ Obligatoire

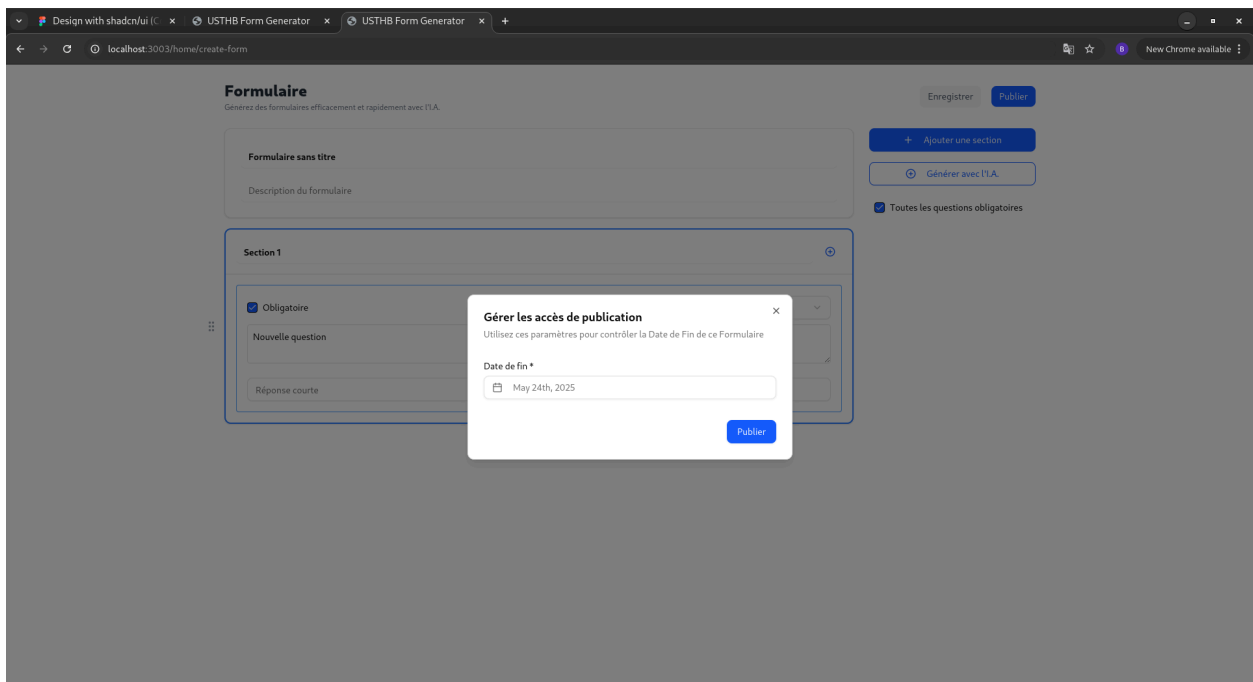
Question courte

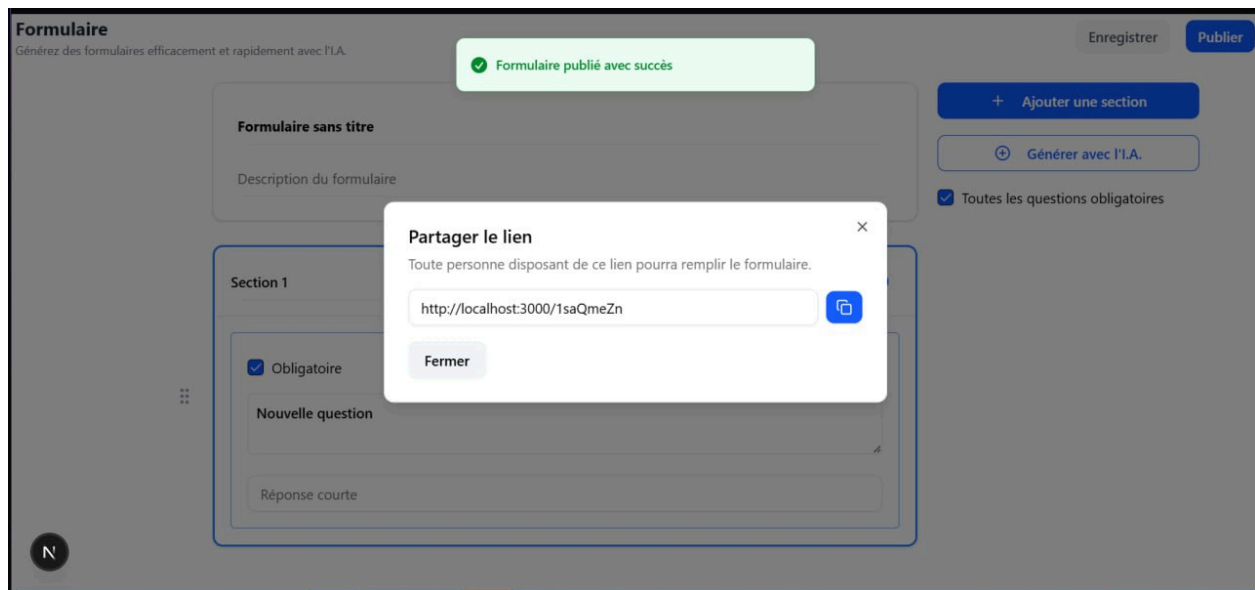
Nouvelle question

Réponse courte

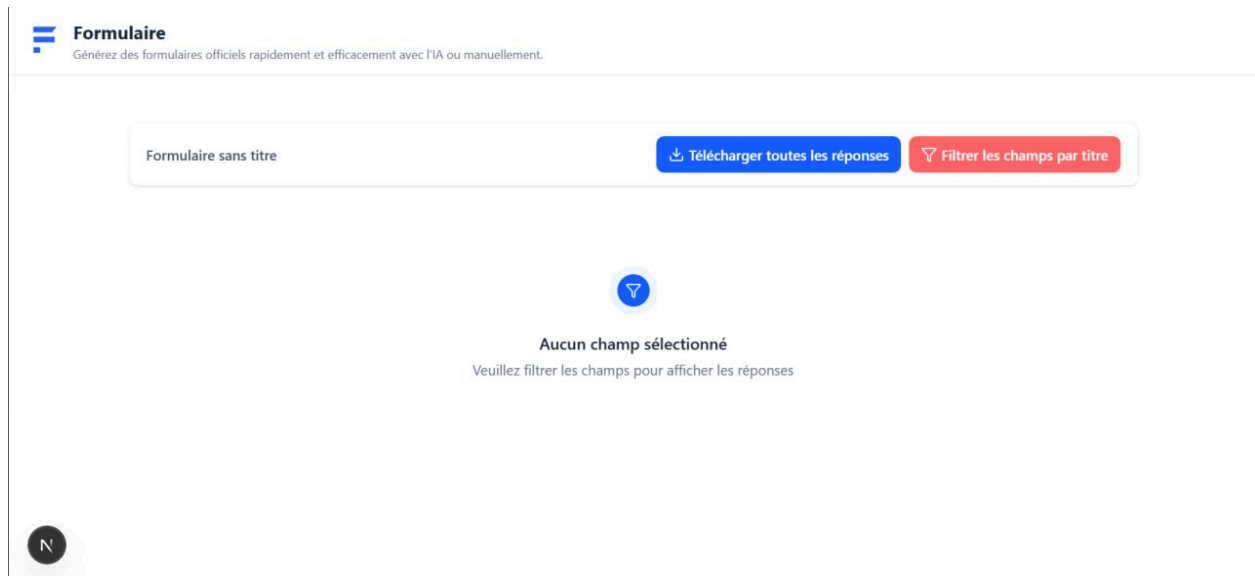
ici c'est les mêmes fonctionnalités pour avec AI et sans AI

Choisir une date pour publier un formulaire

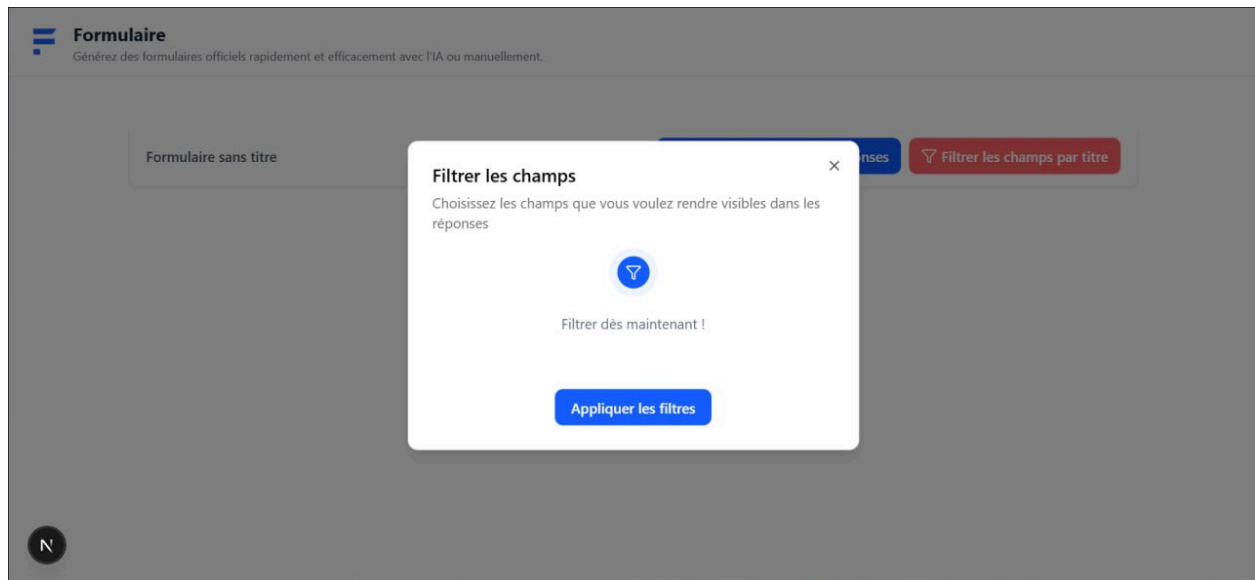




## Page des Reponses



## Filtrage des reponses



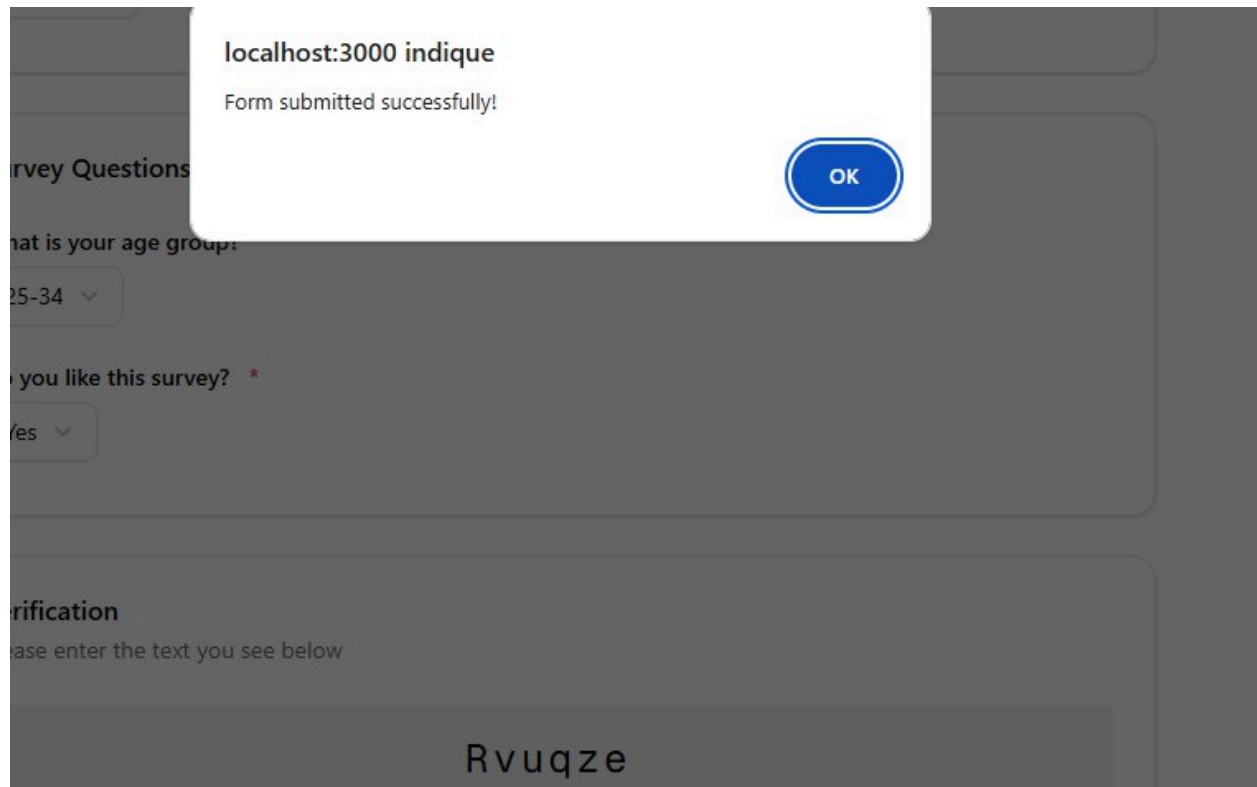
## Formulaire user

**Survey Form 7**  
Please complete the form below

**Personal Information**  
**What is your full name? \***  
  
**What is your gender? \***

**Survey Questions**  
**What is your age group? \***

Après la soumission



The image shows a survey form with a success message overlay. The overlay is a white box with a blue border and a blue 'OK' button. The text inside the overlay reads: 'localhost:3000 indique' and 'Form submitted successfully!'. The background is a dark gray survey form with the following visible text: 'Survey Questions', 'What is your age group:', '25-34', 'Do you like this survey? \*', 'Yes', 'Verification', 'Please enter the text you see below', and 'Rvuqze'.

localhost:3000 indique  
Form submitted successfully!

OK

Survey Questions

What is your age group:

25-34

Do you like this survey? \*

Yes

Verification

Please enter the text you see below

Rvuqze

Survey Questions

What is your age group? \*

25-34

Do you like this survey? \*

Select an option

Verification

Please enter the text you see below

Rvuqze

Type the code you see above

Submit

Reset Form

localhost:3000 indique  
CAPTCHA incorrect. Please try again.

OK

Survey Questions

What is your age group?

18-24

Do you like this survey? \*

Yes

Verification

Please enter the text you see below

aAr8QP

aAr8Qp

localhost:3000 indique  
Please fill in all required fields.

OK

Survey Questions

What is your age group?

18-24

### Survey Form 7

Please complete the form below


#### Personal Information

What is your full name? \*

What is your gender? \*

#### Survey Questions

## Espace administrateurs



Tous les Formulaire

### Administrateur

Bienvenu dans votre espace !

Le formulaire	Telecharger
FORMULAIRE DES ETUDIANTS	<a href="#">Télécharger toutes les réponses</a>
FORMULAIRE DES ETUDIANTS	<a href="#">Télécharger toutes les réponses</a>
FORMULAIRE DES ETUDIANTS	<a href="#">Télécharger toutes les réponses</a>



## Conclusion

Formulaire USTHB représente une solution innovante pour simplifier et automatiser la création de formulaires au sein de l'université. En combinant une interface intuitive avec la puissance de l'intelligence artificielle, notre plateforme permet:

- Une réduction significative du temps de création
- Une standardisation des formulaires
- Une expérience utilisateur améliorée pour tous les acteurs
- Une gestion centralisée et sécurisée des données

Les technologies choisies (React, Node.js, MongoDB et API IA) offrent une base solide pour un système performant, évolutif et facile à maintenir. Les tests utilisateurs initiaux montrent une adoption rapide et une satisfaction élevée parmi les administrateurs.