

# 1. Projet Frontend – React & TypeScript (sans back-end)

## Objectif du projet

Vous allez développer **une application web en React avec TypeScript**, à partir d'une maquette fournie.

Ce projet a 4 objectifs principaux :

1. **Comprendre et structurer une application React**
2. **Manipuler des données et des composants avec TypeScript**
3. **Gérer des interactions utilisateur sans back-end**
4. **Travailler comme une équipe de développeur·euses en mission**

 Pas d'API distante

 Pas de back-end

 Pas d'authentification

 Données locales uniquement (JSON)

## Ce que vous allez construire

Une application solidaire autour **d'un seul thème** (choisi par le groupe) :

-  **Adaopte** : adoption d'animaux abandonnés
- OU
-  **Adaence** : recréer du lien avec les personnes âgées

Dans les deux cas, l'application comporte :

- une page d'accueil
- une page de recherche / liste
- des filtres
- des composants réutilisables
- une réflexion de modélisation des données
- une base de données **conceptuelle** (sans serveur)

## 2. Étape 1 – Initialisation du projet React

 Avant de coder des fonctionnalités, vous devez poser des bases saines.

### Obligatoire

- Projet créé avec **React + TypeScript**

- Arborescence claire :
  - pages
  - components
  - data
  - types
- L'application démarre sans erreur

💡 À ce stade :

- Pas de logique métier
- Pas de CSS avancé
- On valide que le projet est **propre et compréhensible**

### 3. 📄 Étape 2 – Structure des pages & composants

Vous allez maintenant construire la **structure de l'application**.

#### Obligatoire

- Une **Navbar** sous forme de composant
- Au moins **2 pages** :
  - Accueil
  - Page de liste
- Mise en place de la navigation entre les pages
- Découpage en composants :
  - Card
  - List
  - Filter
  - Button

👉 Objectif clé :

#### Comprendre la responsabilité de chaque composant

### 4. ⚙️ Étape 3 – Mise en page & styles

Vous appliquez maintenant les styles à votre application.

#### Obligatoire

- Respect de la maquette fournie

- Mise en page avec flexbox / grid
- Utilisation des variables CSS fournies
- Lisibilité sur desktop

## Bonus

- Responsive (mobile / tablette)

⚠ Le design ne doit **jamais casser la structure** ni la logique React.

## 5. 🛡 Étape 4 – Données & TypeScript

Vous allez travailler avec des données **typées**.

### Obligatoire

- Données stockées dans un fichier JSON local
- Création des **types TypeScript** correspondants
- Utilisation des types dans les composants
- Affichage dynamique des données

👉 Exemple :

- Animal
- Senior
- FilterOptions

💡 Objectif :

Comprendre pourquoi TypeScript **sécurise le code**.

## 6. 🗃 Étape 5 – Interactions & logique métier

Vous allez maintenant rendre l'application interactive.

### Obligatoire

- Gestion des événements utilisateur (clics)
- Filtrage des données
- Gestion des paramètres d'URL (search params)
- Utilisation du localStorage
- Fonctions claires et réutilisables
- Utilisation des conditions et boucles

👉 Exemple :

- Cliquer sur “Rencontrer” affiche un console.log()
- Filtrer par type ou localisation
- Conserver un filtre actif après rechargement

## Bonus

- Pagination

## 7. Étape 6 – Accessibilité & bonnes pratiques

### Obligatoire

- HTML sémantique dans les composants
- Images avec alt pertinents
- Boutons accessibles
- Structure claire des titres
- Pas de div “fourre-tout”

## . Étape 7 – Modélisation de la base de données

Même sans back-end, vous devez **penser vos données comme des pros**.

### Obligatoire

- Un schéma de base de données simple
- Présence de :
  - tables
  - colonnes et types
  - primary keys
  - foreign keys
- Un fichier .sql contenant :
  - CREATE TABLE
  - INSERT INTO

 Le type de base doit être précisé en commentaire :

```
# sqlite
```

## 9. Organisation, Git & travail d'équipe

Vous travaillez **en conditions réelles**.

## **Obligatoire**

- Stand-up tous les matins à partir du jour 2
- Groupes de 4 maximum
- Répartition claire des rôles
- Git utilisé quotidiennement :
  - commits réguliers
  - messages en anglais
  - conventions respectées
- Un document d'organisation :
  - planning par jour
  - temps de pause
  - répartition des tâches

## 10. Ce qui n'est PAS demandé

### **Ce qui n'est PAS demandé**

- Pas d'API
- Pas de back-end
- Pas de login / compte
- Pas de framework CSS complexe
- Pas d'images générées par IA
- Pas de fonctionnalités hors scope

 Ce projet évalue la **qualité**, pas la quantité.

## 11. Rendu attendu

- Un lien GitHub vers le projet React
- Un fichier de schéma de base de données
- Un fichier .sql
- Un projet qui démarre sans erreur

## 12. Thème A1 - Adaopte

### **Instructions**

Vous devez reproduire le site **Adaopte** 

**Contexte :** Adaopte est une plateforme solidaire dédiée à l'adoption responsable, pour offrir une seconde chance aux animaux abandonnés et sensibiliser au bien-être animal.

### Ce que vous devez développer :

- Une navbar
- La page d'**Accueil**
  - Le bouton "Rechercher" amène vers la page "**J'adopte**" avec les critères sélectionnés
  - Le bouton "Voir tous les animaux" amène vers la page "J'adopte"  
Aucun filtre n'est donc appliqué
- La page "**J'adopte**"
  - Liste tous les animaux abandonnés à adopter
  - Il est possible de filtrer par "Type d'animal" et "Localisation"
  - Le clic sur le bouton **Rencontrer** devra (aujourd'hui) simplement faire un `console.log()` du Prénom de l'animal et son Type (chien, chat, rongeur, etc)
  - BONUS : Gérer une pagination
- BONUS : La page "**Devenir bénévole**" (formulaire qui envoie un email)
- Le lien "**Nous contacter**" dans le footer est un `mailto:`
- Une gestion d'erreur sur les formulaires lorsqu'ils ne sont pas correctement remplis

### Vous n'avez pas à développer les pages suivantes :

- De création de compte
- De connexion
- "Faire un don"
- "Guide de l'adoption"
- "FAQ"
- "Conseils d'adoptions"
- "Mentions légales"

**i** La galerie de photos est fixe et ne varie pas.

---

Si vous n'avez pas d'instructions sur certaines pages ou liens, c'est que nous partons sur le principe que le développement n'est pas obligatoire. Vous êtes libres de vous les approprier.

**💡** Nous vous recommandons fortement de **réfléchir à la conception de votre site avant** de vous lancer pour pouvoir réutiliser des parties de votre code pour créer les différentes pages !

Ne partez pas tout de suite à vouloir personnaliser la police et les couleurs. Le premier objectif dans ce genre de projet est de surtout réussir à placer nos éléments correctement et surtout avoir un code lisible et réutilisable !

## Maquette

### Page 1 : Accueil

**i** Les images utilisées dans les maquettes peuvent être récupérées en cliquant sur les liens dans la **Fiche projet**.

Pour la palette de couleurs, vous les retrouverez dans le chapitre **Ressources**.

### Page 2 : J'adopte (Recherche)

Les champs avec les doubles accolades sont à personnaliser une fois l'intégration terminée.

Contrairement à la page d'accueil, ici, pour l'affichage des animaux, il est recommandé d'utiliser JavaScript.

### (BONUS) Page 3 : Devenir bénévole

Pour rappel, n'hésitez pas à reutiliser des blocs de code que vous avez potentiellement déjà fait pour la page d'Accueil !

## Couleurs utilisées

```
:root {  
    --primary-color: #ffffaf7;  
    --secondary-color: #ff6d38;  
    --accent-color: #7a78ff;  
    --heading-color: #333;  
    --light-color: #f9f9f9;  
    --dark-color: #333;  
  
    --btn-primary-bg: #333;  
    --btn-secondary-bg: #e74c3c;  
    --btn-accent-bg: #00a652;  
    --btn-primary-hover-bg: #8584ff;
```

```

--btn-secondary-hover-bg: #333;

--font-family-heading: 'Amatic SC', sans-serif;
--border-radius: 8px;

--shadow-sm: 0 2px 5px rgba(0, 0, 0, 0.1);
--shadow-md: 0 3px 10px rgba(0, 0, 0, 0.1);
--shadow-lg: 0 5px 15px rgba(0, 0, 0, 0.1);
--shadow-xl: 0 5px 15px rgba(0, 0, 0, 0.2);

--section-margin: 2rem 0;
--container-width: 90%;

}

```

## Données sur les animaux

N'hésitez pas à t'aider de l'IA générative pour ajouter des animaux dans la liste.

```

[
{
  "name": "Charlie",
  "age": "5 ans",
  "breed": "Pug",
  "type": "Chien",
  "city": "Lille",
  "zipcode": "59000",
  "description": "Charlie est un petit chien calme et affectueux qui adore les balades tranquilles et les caresses sur le canapé.",
  "imageUrl": "/images/charlesdeluvio-K4mSJ7kc0As-unsplash.jpg"
},
{
  "name": "Mia",
  "age": "2 ans",
  "breed": "Chat noir et blanc",
  "type": "Chat",
  "city": "Strasbourg",
  "zipcode": "67000",
  "description": "Mia est une boule de tendresse curieuse et joueuse, toujours prête à ronronner près de vous."
}
]
```

```
        "imageUrl": "/images/manja-vitolic-gKXKBY-C-Dk-unsplash.jpg"
    },
    {
        "name": "Coco",
        "age": "1 an",
        "breed": "Lapin",
        "type": "Lapin",
        "city": "Annecy",
        "zipcode": "74000",
        "description": "Coco est un petit lapin plein d'énergie, très doux et parfait pour un foyer aimant et paisible.",
        "imageUrl": "/images/chan-swan-NKyI19P5IHg-unsplash.jpg"
    },
    {
        "name": "Rex",
        "age": "4 ans",
        "breed": "Chow-chow",
        "type": "Chien",
        "city": "Grenoble",
        "zipcode": "38000",
        "description": "Rex est un chien loyal et majestueux, idéal pour une personne douce et patiente qui aime les câlins silencieux.",
        "imageUrl": "/images/alan-king-KZv7w34tluA-unsplash.jpg"
    },
    {
        "name": "Luna",
        "age": "3 ans",
        "breed": "Chat Roux",
        "type": "Chat",
        "city": "Bordeaux",
        "zipcode": "33000",
        "description": "Luna adore les coins ensoleillés et les siestes à vos côtés. Elle vous offrira tout l'amour d'un regard félin.",
        "imageUrl": "/images/jae-park-7GX5aICb5i4-unsplash.jpg"
    },
    {
```

```
"name": "Biscuit",
"age": "3 mois",
"breed": "Cochon d'Inde",
"type": "Rongeur",
"city": "Tours",
"zipcode": "37000",
"description": "Biscuit est tout petit mais plein de vie ! Il aime les légumes croquants et les instants de douceur en famille.",
"imageUrl": "/images/yosei-g-0VgE3m4MHKM-unsplash.jpg"
},
{
"name": "Rio",
"age": "5 mois",
"breed": "Lapin nain brun",
"type": "Lapin",
"city": "Poitiers",
"zipcode": "86000",
"description": "Rio est discret, mignon comme tout, et adore explorer les petits coins de la maison avec délicatesse.",
"imageUrl": "/images/melanie-kreutz-IFnknR2Mv5o-unsplash.jpg"
},
{
"name": "Ruby",
"age": "2 ans",
"breed": "Chien Samoyed",
"type": "Chien",
"city": "Chamonix",
"zipcode": "74400",
"description": "Ruby est une boule de neige pleine d'amour ! Elle adore les promenades et les câlins par temps frais.",
"imageUrl": "/images/peri-stojnic-5Vr_RVPfbMI-unsplash.jpg"
}
]
```