OK, bro ! Voici un document de conception complet pour le site d’administration qu’on a construit. Je vais structurer ça de manière claire et professionnelle, en détaillant l’architecture, les fonctionnalités, les choix techniques, et les aspects visuels. Comme c’est un doc complet, ça va être un peu long, mais je vais le rendre digeste avec des sections bien définies. Allons-y !

Document de Conception : Site d'Administration "Admin Campus"

1. Introduction

1.1 Objectif

Le site "Admin Campus" est une application web conçue pour permettre aux administrateurs de gérer les bâtiments, salles, et autres infrastructures d’un campus. L’objectif principal est de fournir une interface intuitive et sécurisée pour effectuer des opérations CRUD (Create, Read, Update, Delete) sur les données stockées dans Firebase, avec des fonctionnalités avancées comme des notifications et un accès restreint.

1.2 Portée

---

2. Architecture

2.1 Vue d’ensemble

L’application est une SPA (Single Page Application) construite avec React, intégrant Firebase pour le backend et Tailwind CSS pour le style. Voici les composants principaux :

• Frontend

• Backend

• CI/CD

2.2 Structure des fichiers

```
admin-campus/
%%%   public/
%     %%%   index.html                # Point d'entrée HT
%     %%%   favicon.ico
%%%   src/
%     %%%   components/               # Composants réutil
%     %     %%%   BatimentList.js
%     %     %%%   SalleList.js
%     %     %%%   InfrastructureList.js
%     %     %%%   Sidebar.js
%     %     %%%   Spinner.js
%     %     %%%   BuildingForm.js
%     %     %%%   RoomForm.js
%     %     %%%   InfrastructureForm.js
%     %     %%%   BuildingDetailsModal.js
%     %     %%%   InfrastructureDetailsModal.js
%     %     %%%   ConfirmationModal.js
%     %%%   context/                  # Contextes React
%     %     %%%   AuthContext.js
%     %     %%%   CampusContext.js
%     %%%   pages/                    # Pages principales
%     %     %%%   Batiments.js
%     %     %%%   Salles.js
%     %     %%%   AutreInfrastructures.js
%     %     %%%   Login.js
%     %%%   services/                 # Services Firebase
%     %     %%%   firebase.js
%     %     %%%   FirebaseService.js
%     %     %%%   AuthService.js
%     %%%   App.js                    # Routes et contene
%     %%%   index.js                  # Point d'entrée Re
%     %%%   index.css                 # Styles globaux av
%%%   functions/                      # Cloud Functions p
%     %%%   index.js
%     %%%   package.json
%%%   .github/
%     %%%   workflows/
%           %%%   deploy.yml          # Pipeline CI/CD
%%%   firebase.json                   # Config Firebase
%%%   package.json
```

2.3 Flux de données

1. **Authentification**:
   - L'utilisateur se connecte via `/login` avec Firebase Authentication.
   - Un cookie `__session` est créé après connexion réussie.
   - Une Cloud Function vérifie ce cookie pour chaque requête.
2. **Gestion des données**:

- Les données (bâtiments, salles, infrastructures) sont récupérées depuis Firestore via `CampusContex`
- Les opérations CRUD sont effectuées via `FirebaseService` et mises à jour en temps réel dans l'ét

3. **Notifications**:
   - `react-toastify` affiche des messages après chaque action CRUD réussie ou échouée.

Authentification:

- /login
- __session

Gestion des données:

- CampusContext
- FirebaseService

Notifications:

- react-toastify

---

3. Fonctionnalités

3.1 Authentification

- Page de connexion
- Déconnexion
- Accès privé

3.2 Gestion des entités

- Bâtiments
  - Ajout, modification, suppression avec image, nom, et description.
  - Liste scrollable sous forme de grille.
  - Détails dans une modale avec liste des salles associées.
- Salles
  - Ajout, modification, suppression avec image, nom, capacité, description, et lien vers un bâtiment.
  - Liste scrollable avec aperçu horizontal.
- Autres Infrastructures
  - Ajout, modification, suppression avec image, nom, et description.
  - Liste scrollable sous forme de grille.
  - Détails dans une modale.

3.3 Interface utilisateur

- Sidebar
- Recherche

• Notifications
• Spinner

3.4 Déploiement

• CI/CD
• Hébergement

---

4. Choix techniques

4.1 Frontend

• React
• React RouterOutlet
• Tailwind CSS 4.0
• react-toastify

4.2 Backend

• Firebase Firestore
• Firebase Authentication
• Firebase Hosting
• Cloud Functions

4.3 Déploiement

• GitHub Actions

4.4 Police et design

• Poppinsindex.css

---

5. Conception visuelle

5.1 Palette de couleurs

• Primaireindigo-500indigo-600
• Secondairegray-900gray-100
• Erreurred-500red-600
• Succèsreact-toastify

5.2 Typographie

• Police
• Tailles
  • Titres `text-2xl`(Sidebar) `text-xl`(Listes).
  • Texte `text-sm` `text-lg`selon contexte.
• text-2xltext-xl
• text-smtext-lg

5.3 Layout

• Sidebar
• Contenuml-64max-h-[calc(100vh-175px)]
• Modales

5.4 Effets

• Hovertransition-colorshover:shadow-md
• Notificationsreact-toastify

---

6. Sécurité

• Authentification
• Accès privé__session
• Firestore Rules

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```


```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```


---

7. Pipeline CI/CD

7.1 Workflow

• Triggermain
• Étapes
    1. Checkout du code.
    2. Installation de Node.js et dépendances.
    3. Build (`npm run build`).
    4. Déploiement sur Firebase Hosting et Functions.

1.
2.
3: npm run build
4.

7.2 Fichier `deploy.yml`

```
name: Deploy to Firebase Hosting
on:
  push:
    branches:
      - main
jobs:
  build-and-deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
        with:
          node-version: '18'
      - run: npm install
      - run: npm run build
      - run: |
          npm install -g firebase-tools
          firebase deploy --only hosting,functions --token ${{ secrets.FIREBASE_TOKEN }}
```

---

8. Dépendances

8.1 `package.json`

```
{
  "dependencies": {
    "firebase": "^9.6.1",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-icons": "^4.7.1",
    "react-router-dom": "^6.8.1",
    "react-toastify": "^9.1.1"
  },
  "devDependencies": {
    "tailwindcss": "^4.0.0-alpha.1"
  }
}
```

8.2 `functions/package.json`

```
{
  "dependencies": {
    "firebase-admin": "^11.0.0",
    "firebase-functions": "^4.0.0",
    "express": "^4.17.1",
    "cookie-parser": "^1.4.5"
  }
}
```

---

9. Instructions de déploiement

1. Initialiser Firebase `firebase init hosting`
   `firebase init functions`
2. Configurer secrets GitHub
   - Ajoute `FIREBASE_TOKEN` dans **Settings** > **Secrets**.
3. Pousser su `main` `git push origin main`
4. Vérifier https://<ton-projet>.web.app

```
firebase init hosting
firebase init functions
```

- FIREBASE_TOKEN Settings > Secrets

```
git push origin main
```

---

10. Améliorations futures

- Rôles Admin
- Analytics
- Tests
- Style avancé `react-toastify`

Voilà, bro ! Ce doc couvre tout ce qu'on a fait : architecture, fonctionnalités, design, sécurité, et déploiement. Si tu veux que je développe une section ou que je génère un PDF, fais-moi signe ! Ça te va comme ça ?