l'Assurance Maladie
Agir ensemble, protéger chacun
Val-de-Marne

eFrei
PARIS PANTHÉON-ASSAS UNIVERSITÉ

Cooloc

Votre gestionnaire de colocations
CDA 2024 - 2025

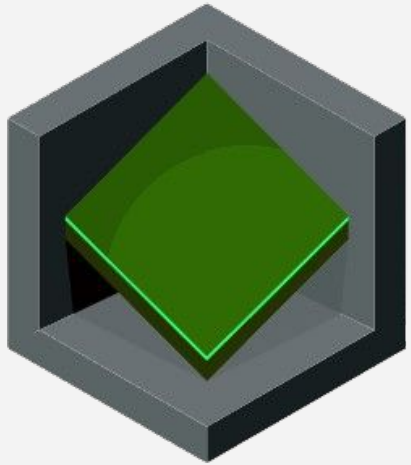
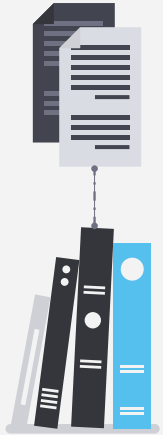
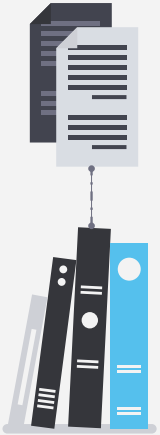


Table des matières



I. Introduction

I. Introduction - Qui suis-je



Data Engineer (stage)

LCL - Villejuif 94800

Concepteur Développeur (alternance)

CPAM du Val-de-Marne - Créteil 94000

2023

2024

2024

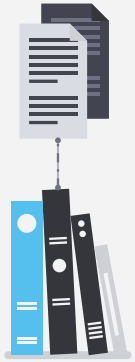
2025

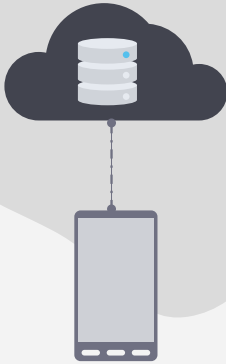
DevOps (stage)

Société Générale - Fontenay-sous-bois 94120

Data Engineer (alternance)

Crédit Agricole Assurances - Paris 75015 - À venir





I. Introduction - Parcours scolaire

Lycée Alexandre Dumas

2019 - 2022
Baccalauréat général
Physique-Chimie & NSI

Efrei Paris

2022 - 2025
Bachelor
Développement web & App

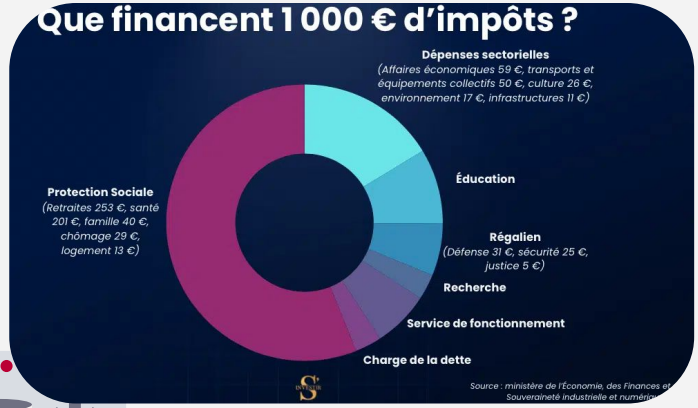
Efrei Paris

2025 - 2027 (a venir)
Mastère
Data Engineering & IA

II. Présentation de l’entreprise

II. Présentation de l'entreprise - CPAM

l'Assurance Maladie
Agir ensemble, protéger chacun
Val-de-Marne



L'Assurance Maladie et la prévention, un engagement historique

1945 — L'ordonnance du 4 octobre 1945 instaure la Sécurité sociale et évoque la « prévention » à 4 reprises

1985 — Organisation de la première campagne de vaccination contre la grippe

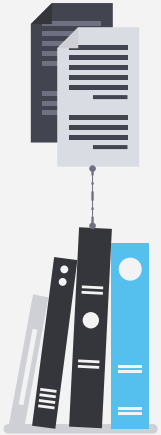
2005 — Lancement du programme M'T dents, qui sera étendu à de nouvelles classes d'âge en 2018

2008 — Déploiement du service sophia, pour améliorer le suivi de patients diabétiques

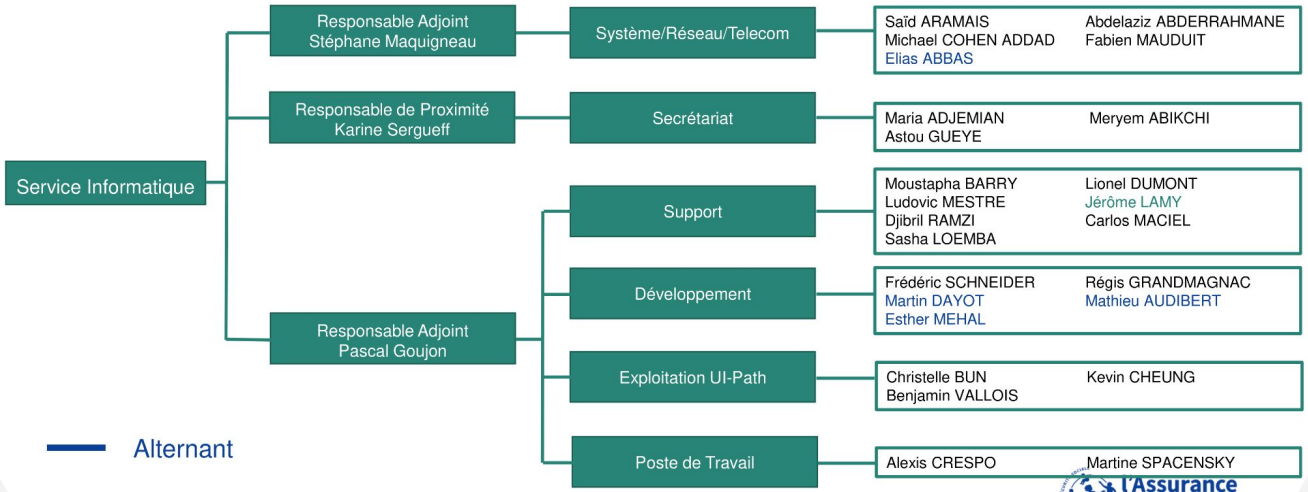
2020 — Mise en place d'actions d' « aller vers » lors de la vaccination anti-Covid

l'Assurance Maladie
Agir ensemble protéger chacun

II. Présentation de l'entreprise - Organigramme

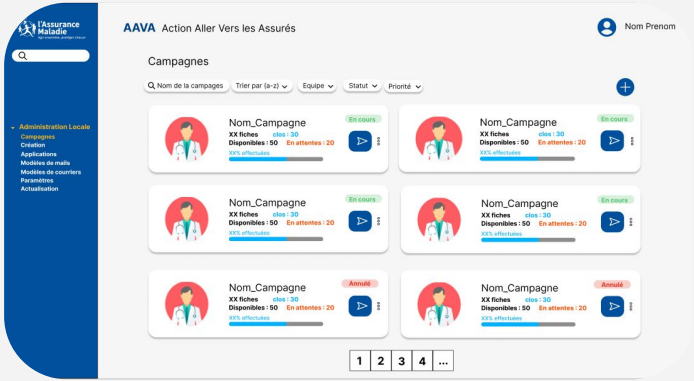


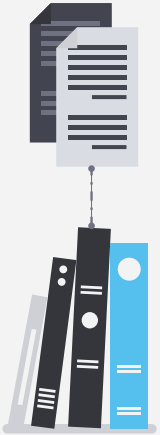
ANNEXE ETAT DES LIEUX – ORGANIGRAMME

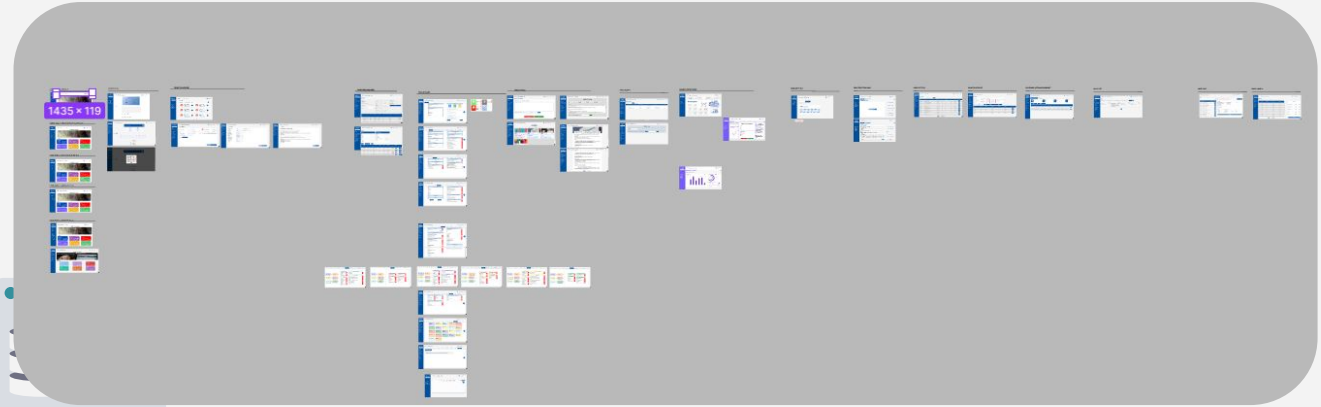


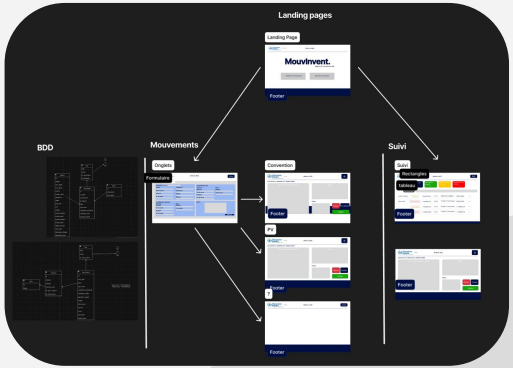
II. Présentation de l'entreprise - Mes missions

AAVA Action Aller Vers les Assurés

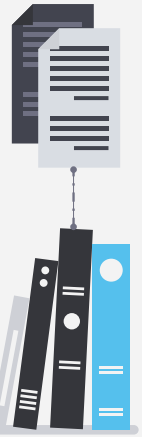








II. Présentation de l'entreprise - Mes missions



GitLab

php

Symfony

MySQL®

III. Présentation de Cooloc

III. Présentation de Cooloc - Contexte

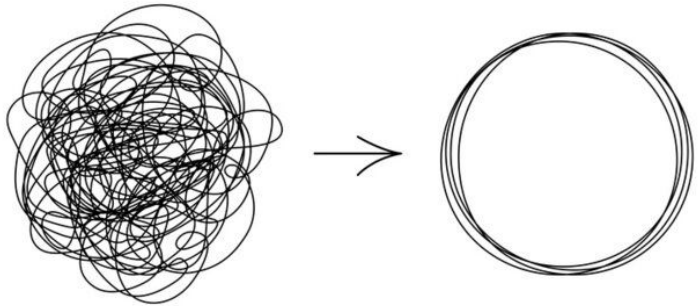
Précarité étudiante en 2023

des étudiants ne mangent pas à leur faim

soit 28% des boursiers et 16% des non-boursiers.

Source : Bouge ton Crous, la consultation étudiante de La Fage, 2024.

III. Présentation de Cooloc - Problématique



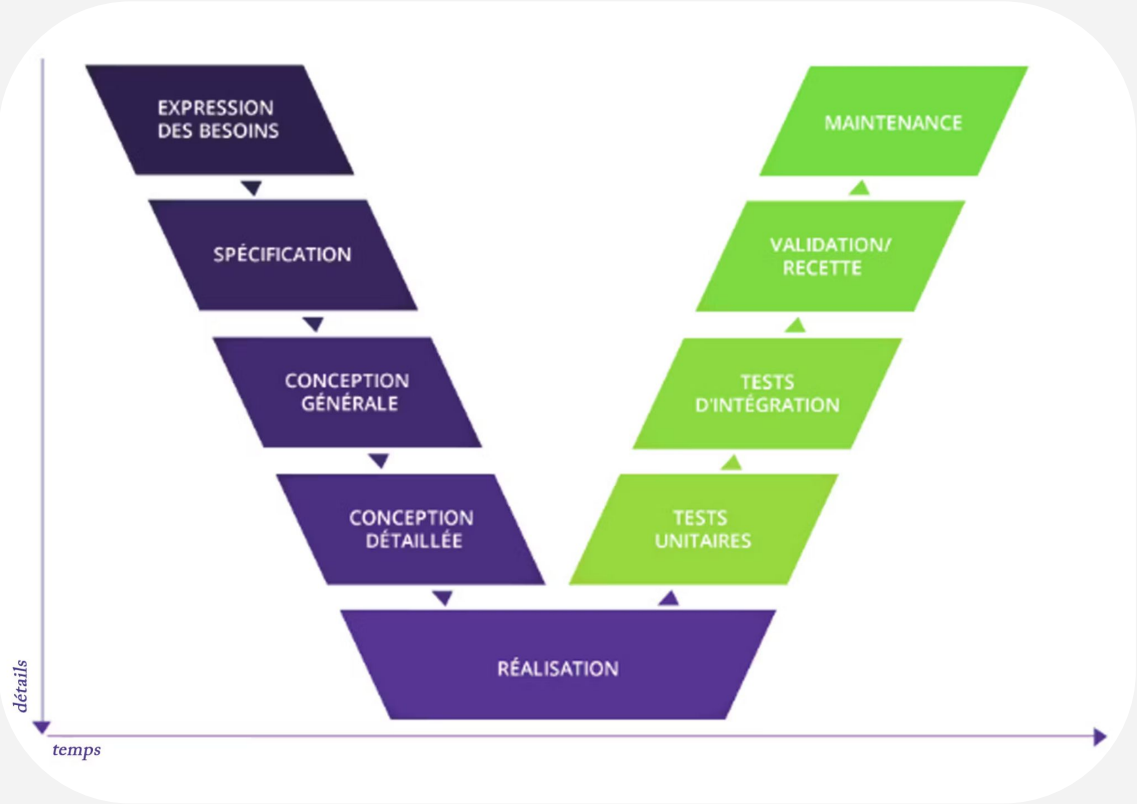
III. Présentation de Cooloc - Objectifs

+Simple

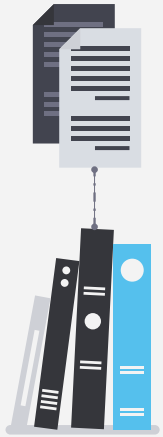
III. Présentation de Cooloc - Public cible

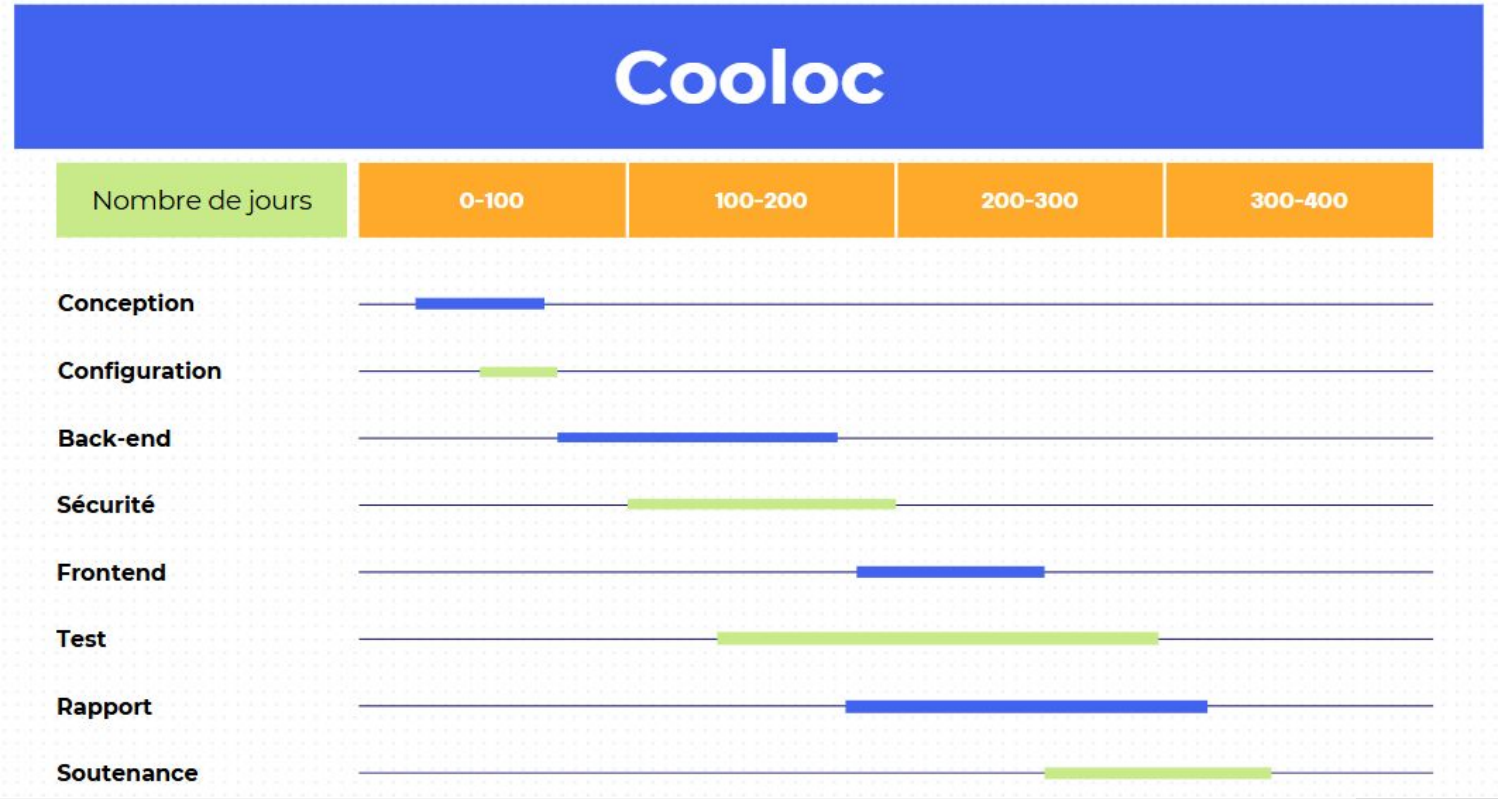
IV. Gestion du projet

IV. Gestion du projet - Méthode

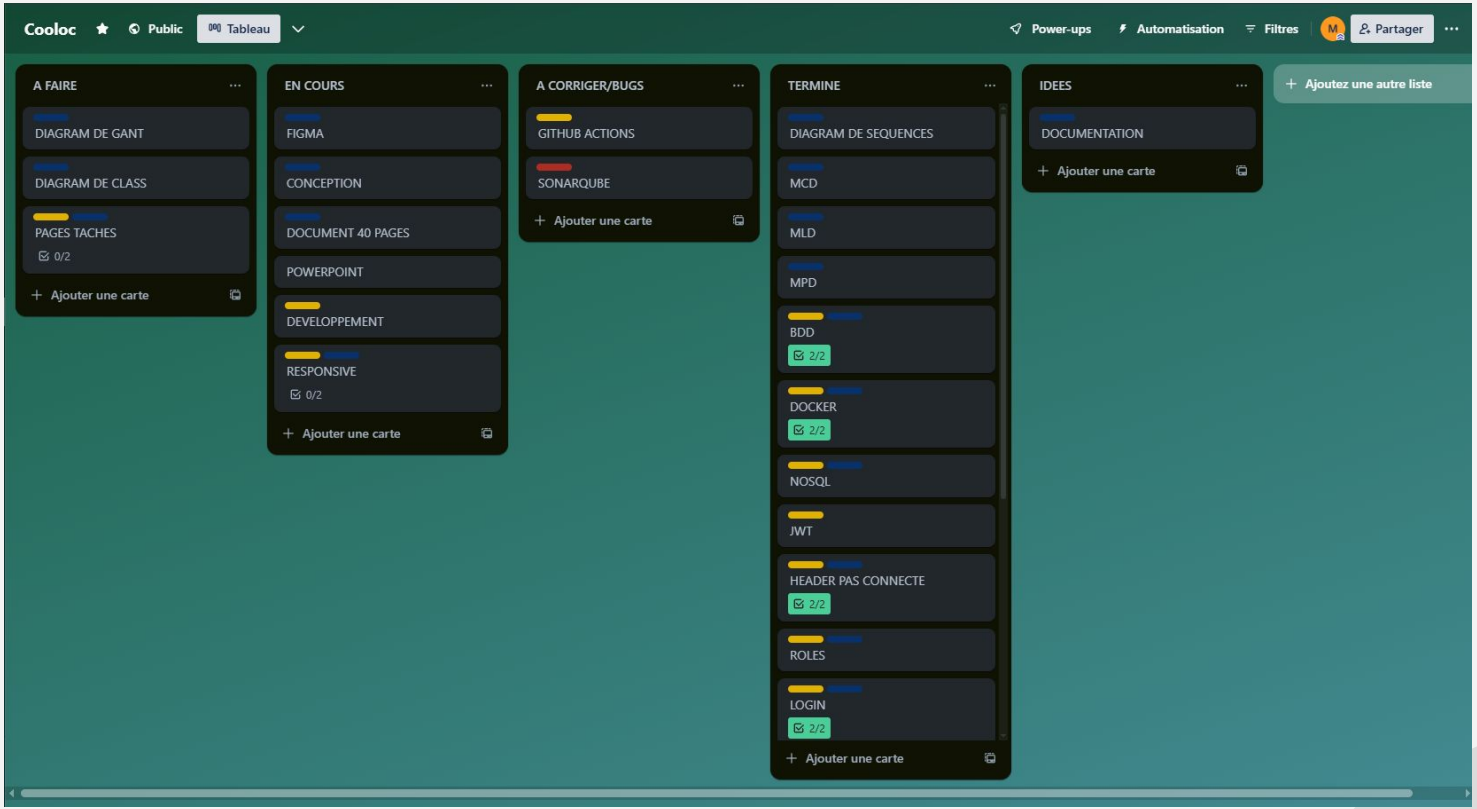


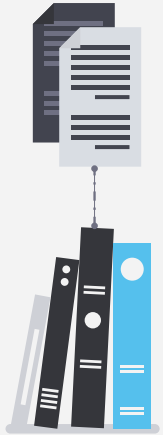
IV Gestion du projet - Planning des tâches



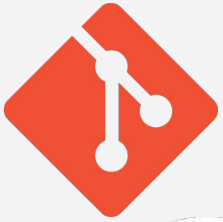


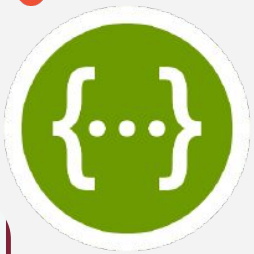
IV. Gestion du projet - Gestion des tâches

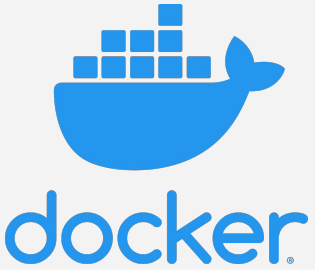


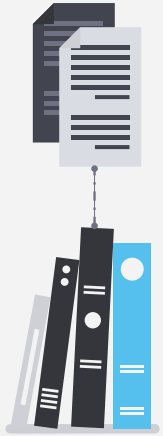


IV. Gestion du projet - Outils

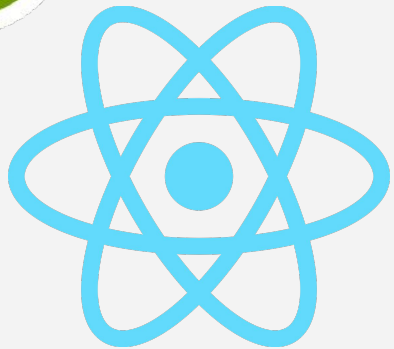


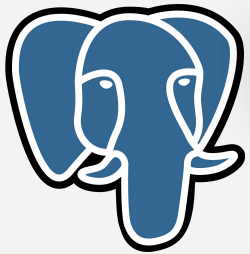






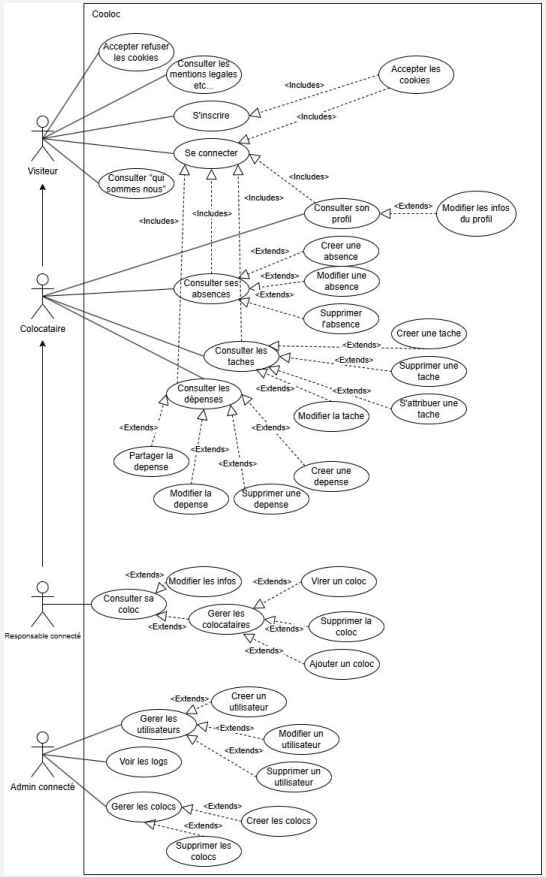
PYTHON
HTTP
MODULE



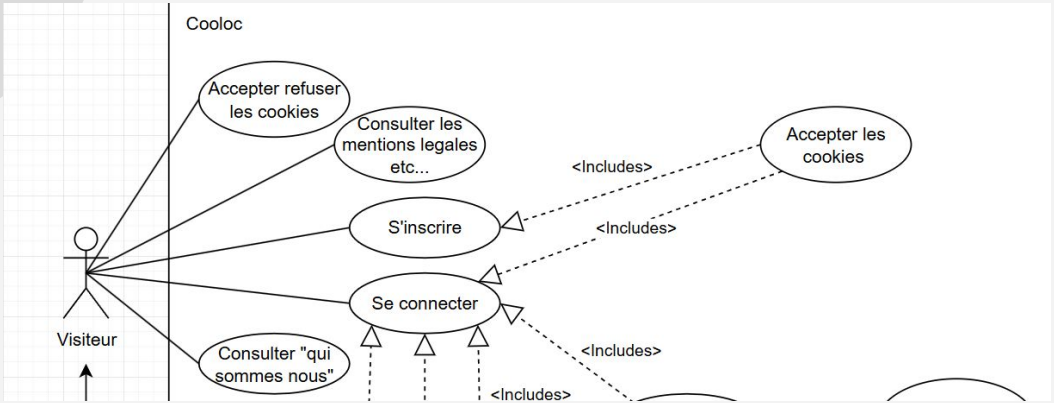


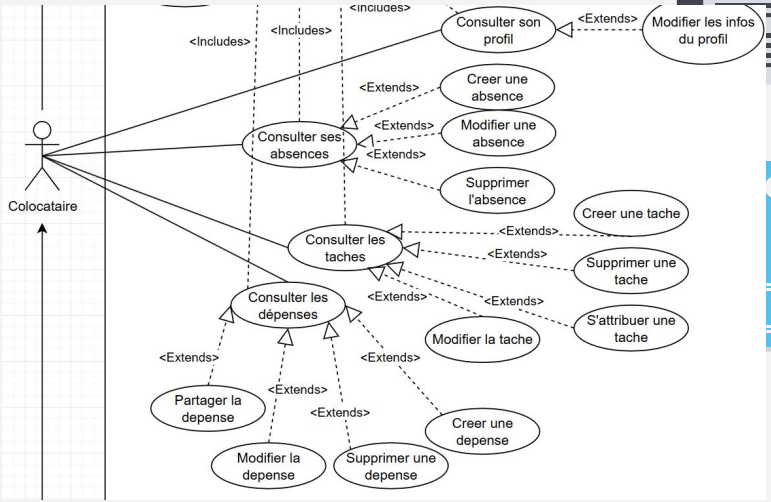
V. Conception de Cooloc

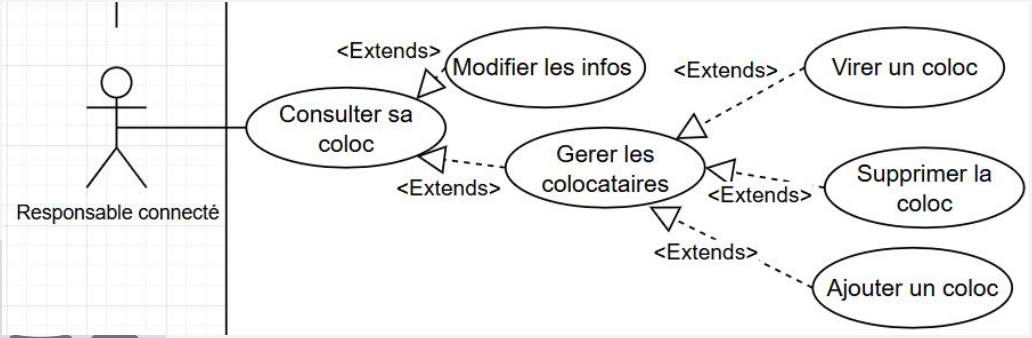
V. Conception de Cooloc - Use Case

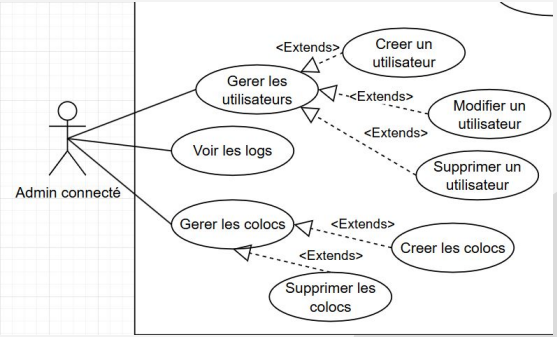


V. Conception de Cooloc – Use Case

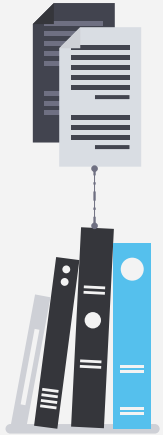


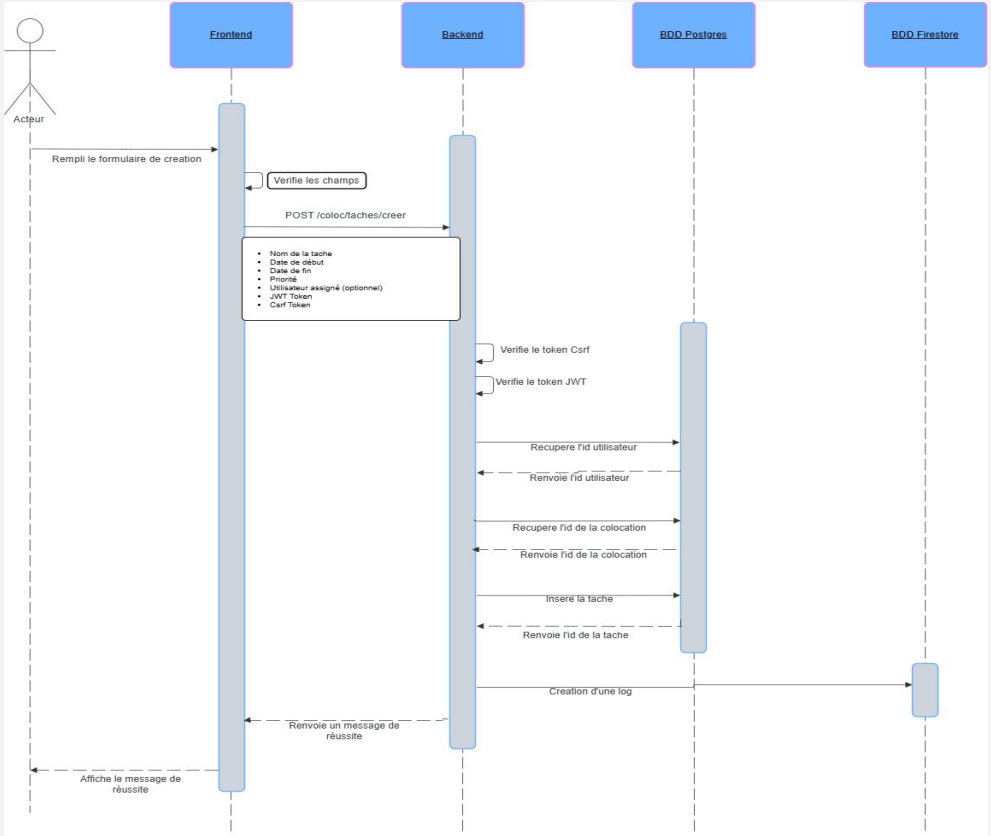




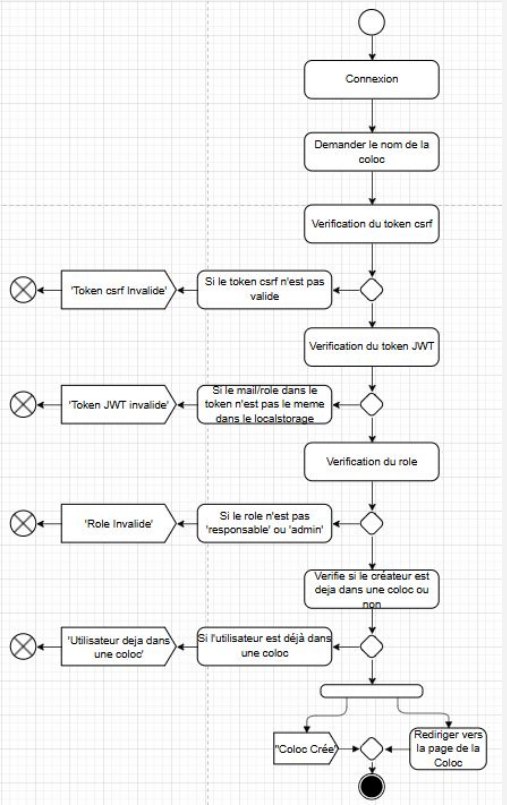


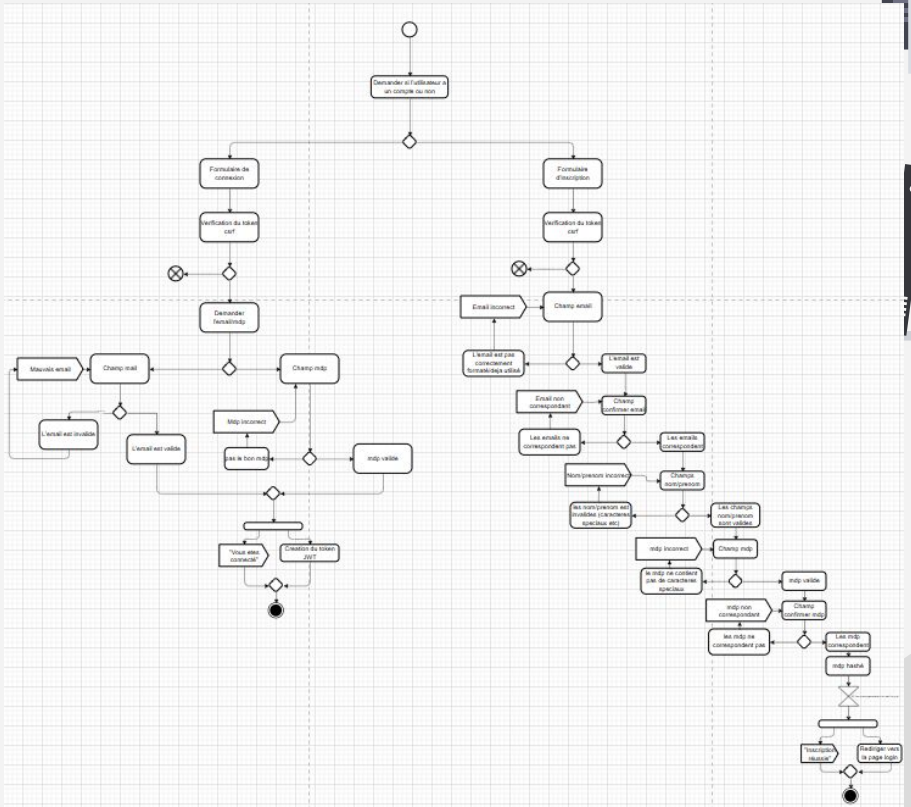
V. Conception de Cooloc – Séquence



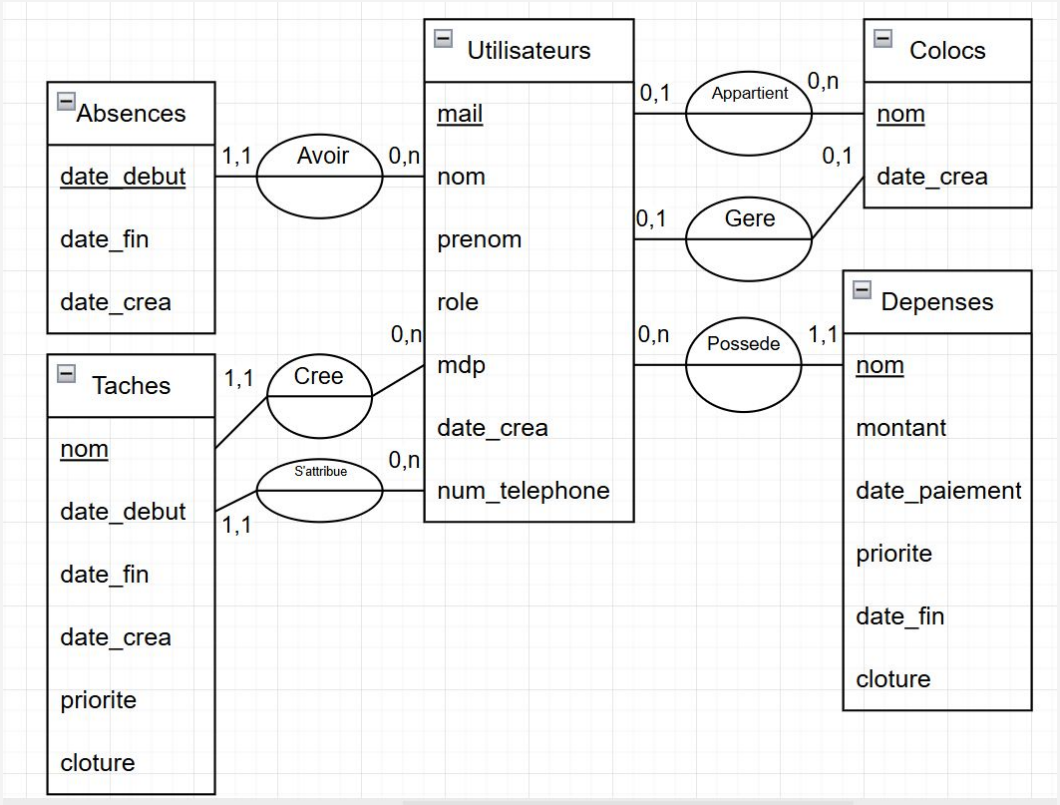


V. Conception de Cooloc - Activité

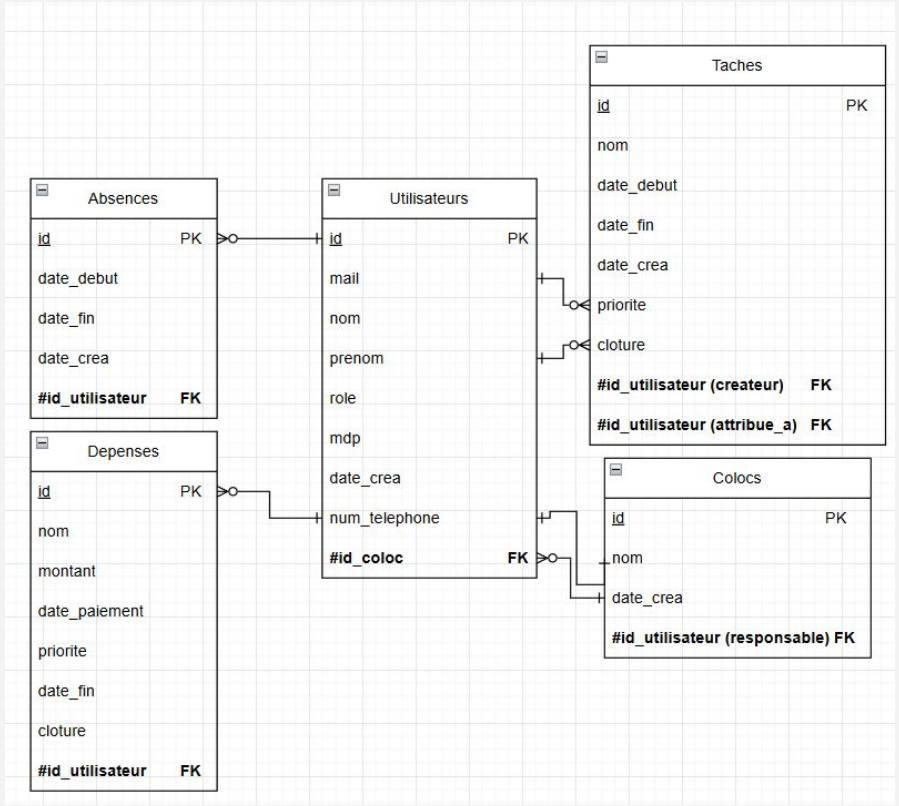




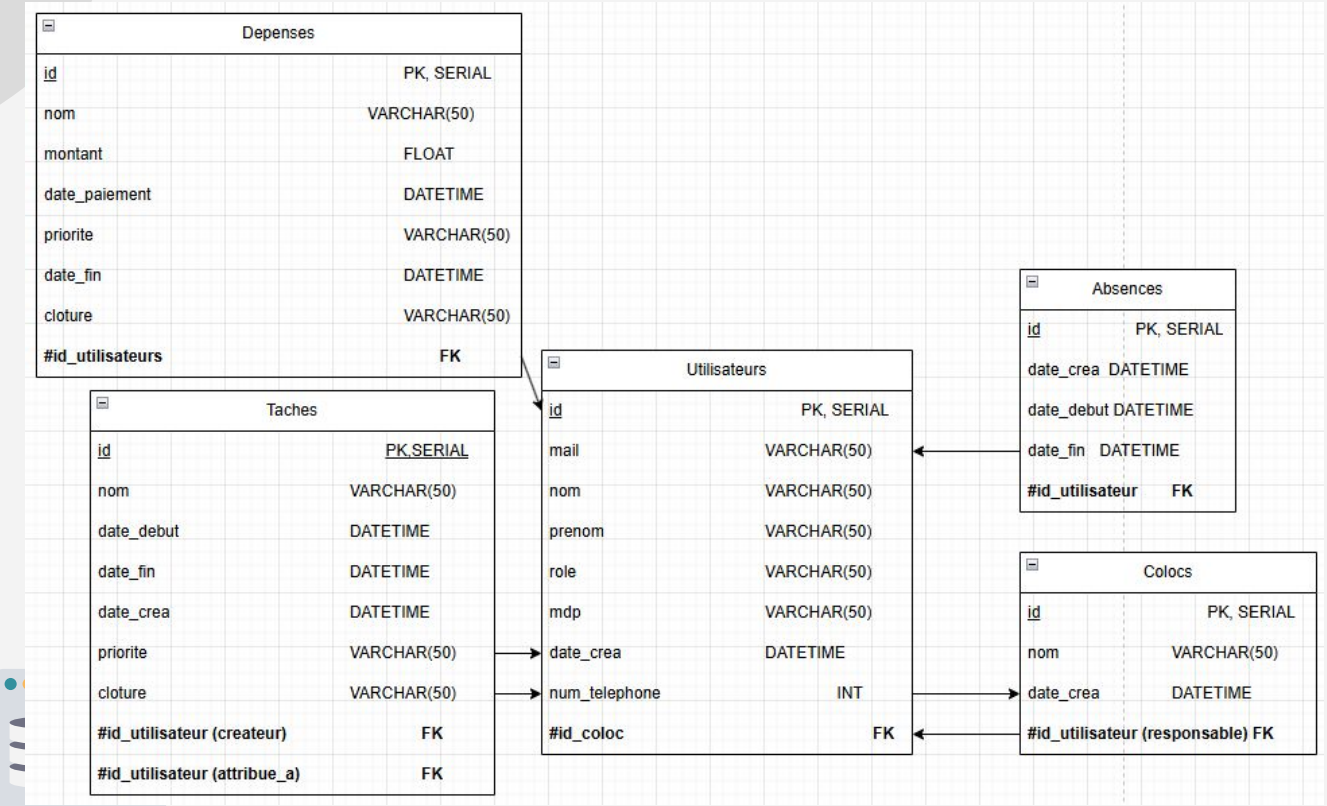
V. Conception de Cooloc – MCD



V. Conception de Cooloc – MLD



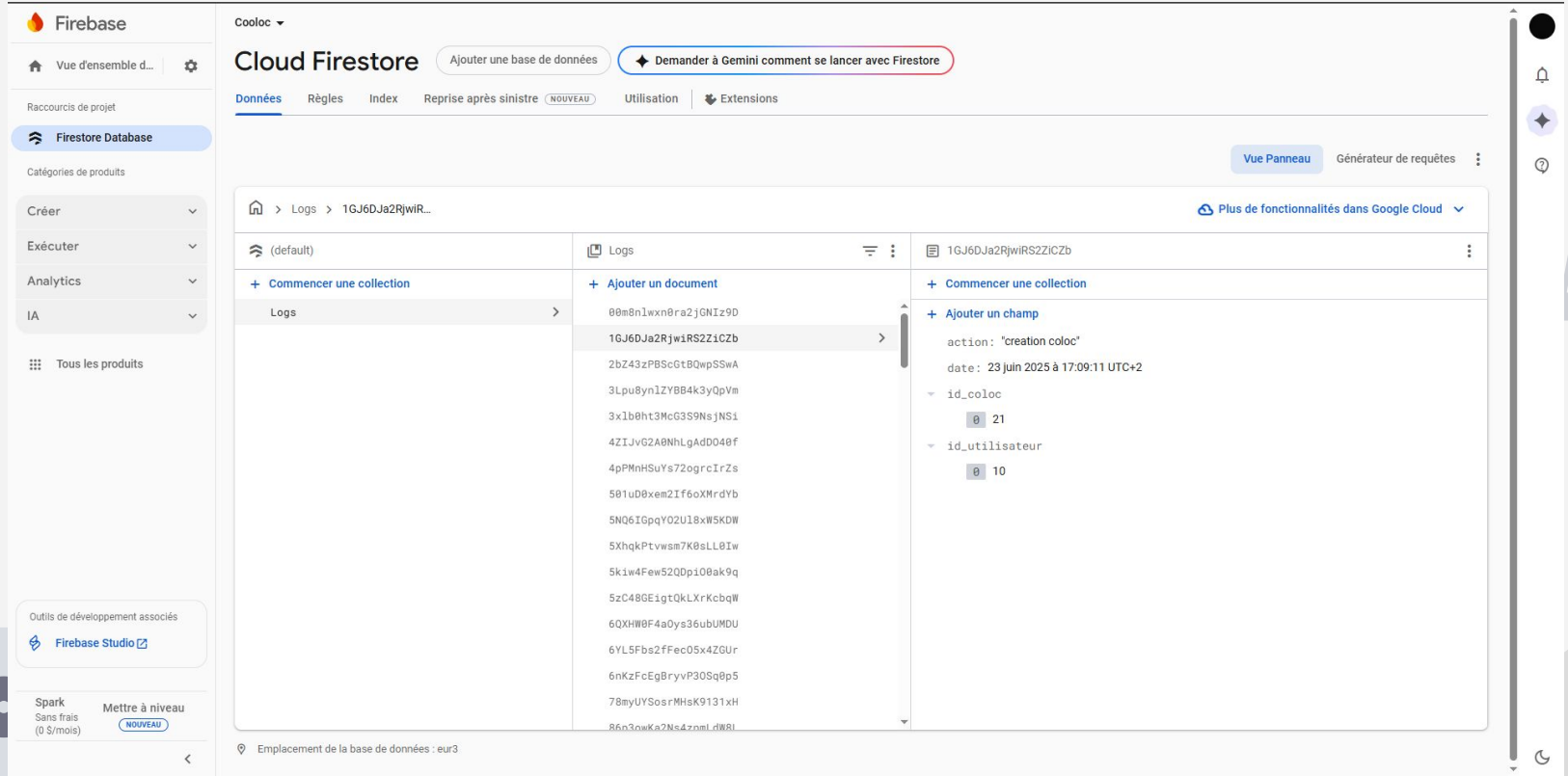
V. Conception de Cooloc – MPD



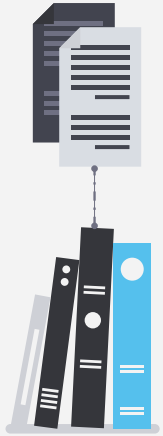
backups > back > bdd.sql

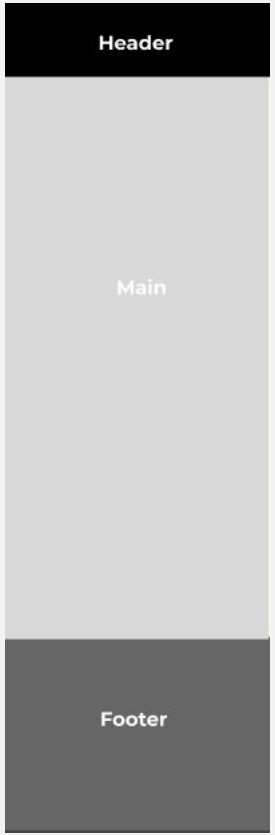
```sql
CREATE TABLE IF NOT EXISTS Colocs (
    id SERIAL PRIMARY KEY,
    nom VARCHAR(50),
    date_crea TIMESTAMP,
    responsable INT,
);

CREATE TABLE IF NOT EXISTS Utilisateurs (
    id SERIAL PRIMARY KEY,
    mail VARCHAR(50) NOT NULL UNIQUE,
    nom VARCHAR(50) NOT NULL,
    prenom VARCHAR(50) NOT NULL,
    role VARCHAR(50) NOT NULL,
    mdp VARCHAR(255) NOT NULL,
    date_creation TIMESTAMP,
    num_telephone VARCHAR(15),
    id_coloc INT REFERENCES Colocs(id),
    CONSTRAINT fk_coloc
        FOREIGN KEY(id_coloc)
            REFERENCES Colocs(id)
);

CREATE TABLE IF NOT EXISTS Absences (
    id SERIAL PRIMARY KEY,
    date_crea TIMESTAMP,
    date_debut TIMESTAMP,
    date_fin TIMESTAMP,
    id_utilisateur INT REFERENCES Utilisateurs(id)
);

CREATE TABLE IF NOT EXISTS Depenses (
    id SERIAL PRIMARY KEY,
    nom VARCHAR(50),
    montant FLOAT,
    date_paiement TIMESTAMP,
```

V. Conception de Cooloc – NoSQL

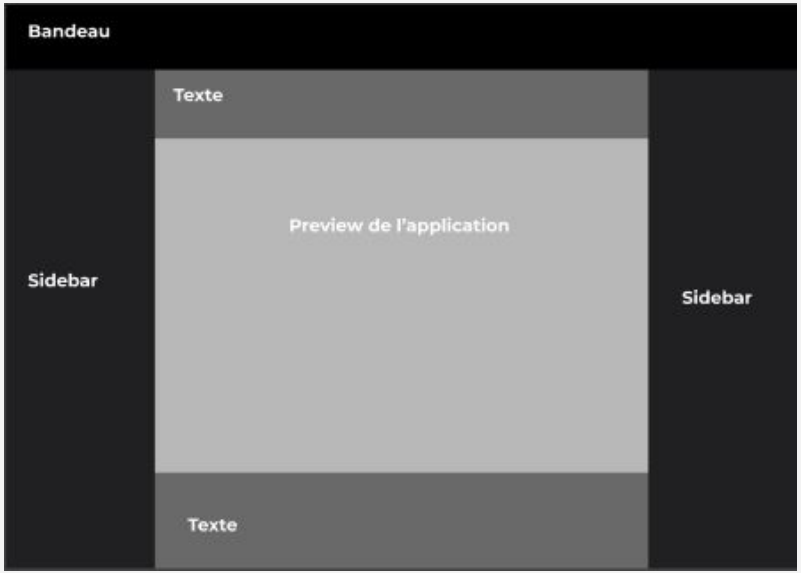


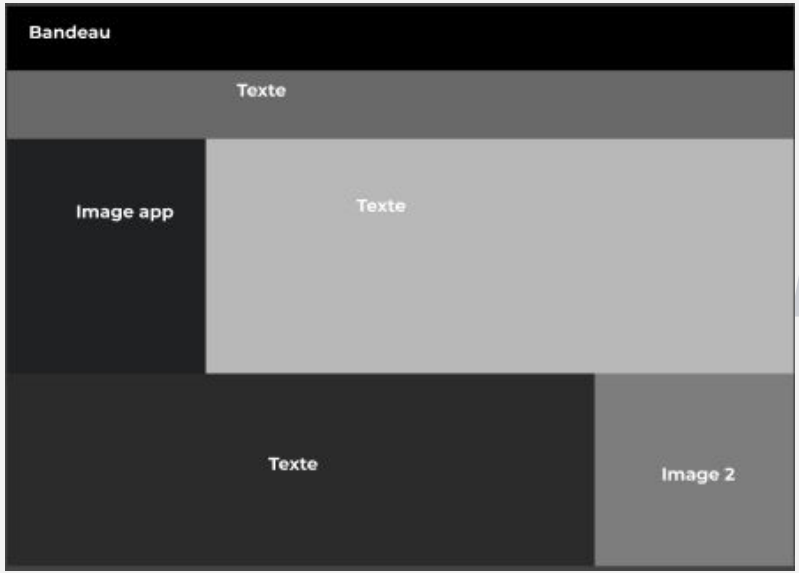
V. Conception - Zoning mobile

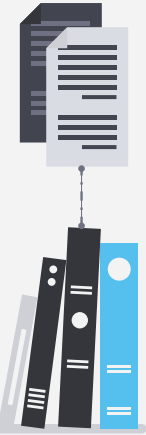




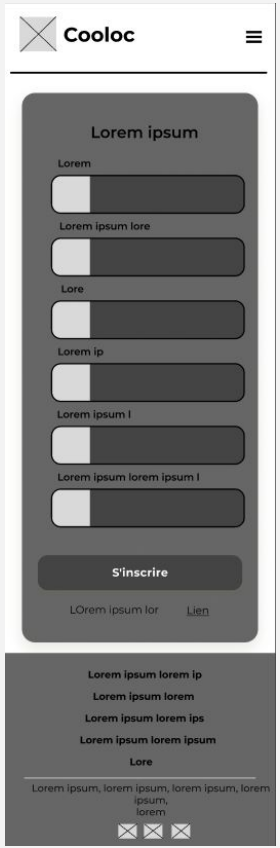
V. Conception – Zoning pc



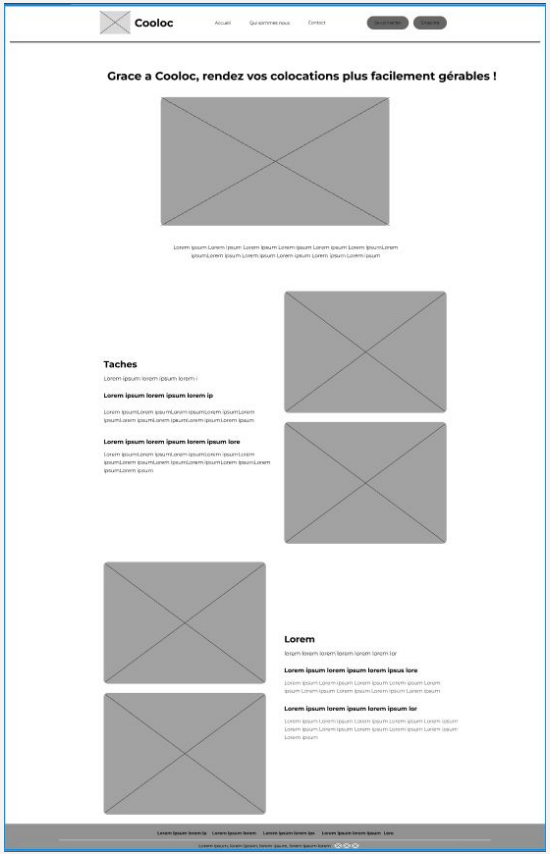


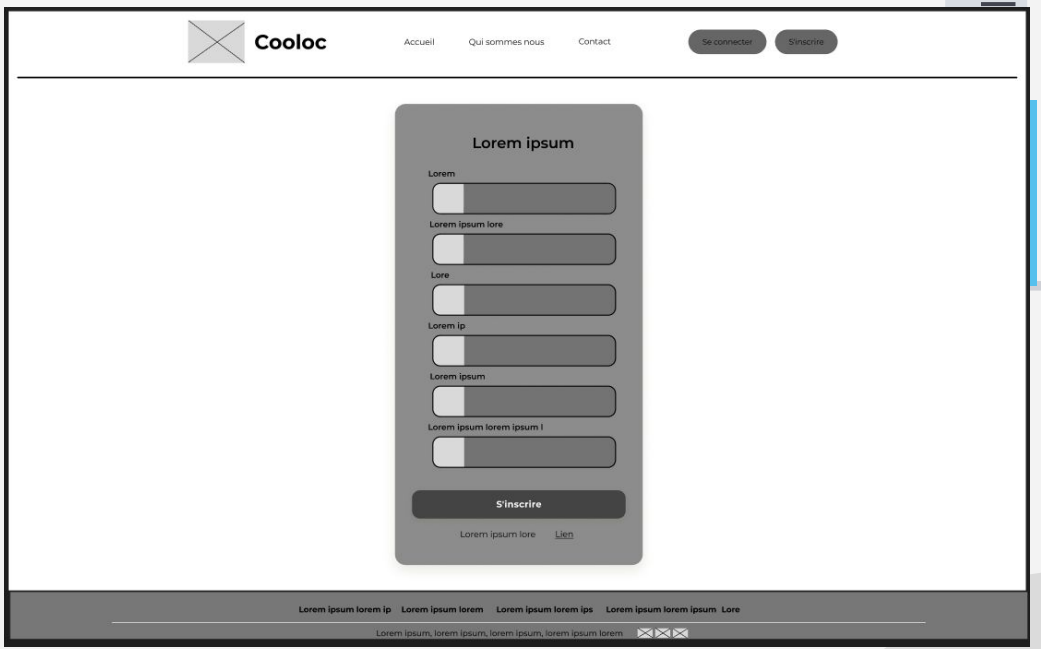


V. Conception - Wireframes mobile



V. Conception - Wireframes pc





V. Conception – Charte graphique

Couleurs

Logo & Polices

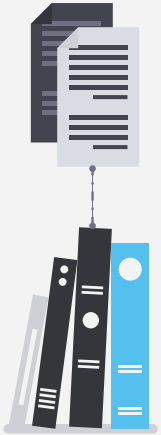
Cooloc utilise Montserrat, Lorem ipsum 123456789

Cooloc utilise Montserrat, Lorem ipsum 123456789

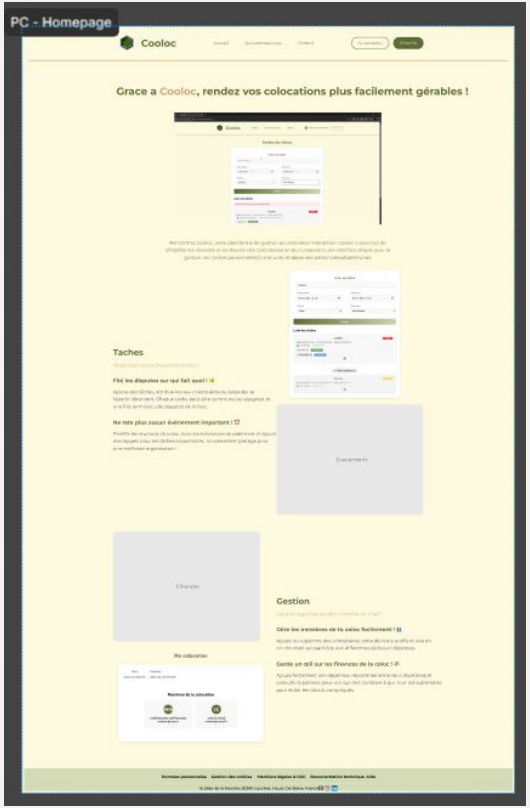
Cooloc utilise Montserrat, Lorem ipsum 123456789

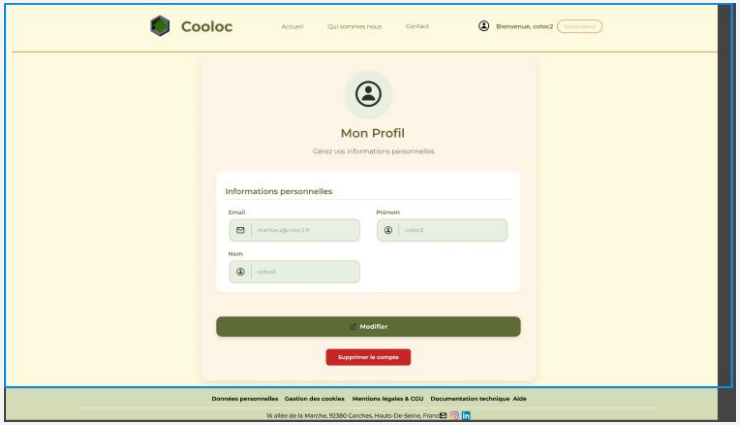
Cooloc utilise Montserrat, Lorem ipsum 123456789

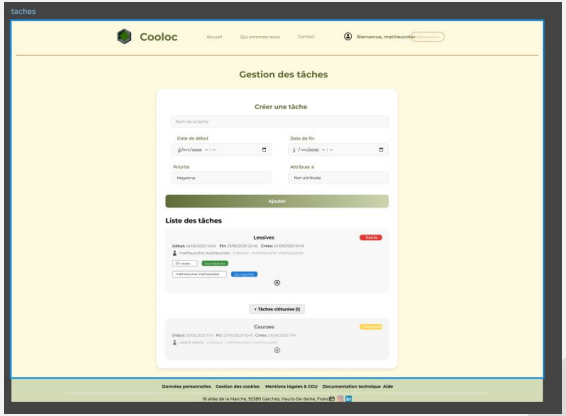
V. Conception - Maquettes mobile

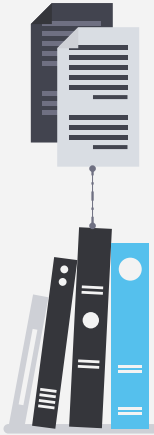


V. Conception - Maquettes pc

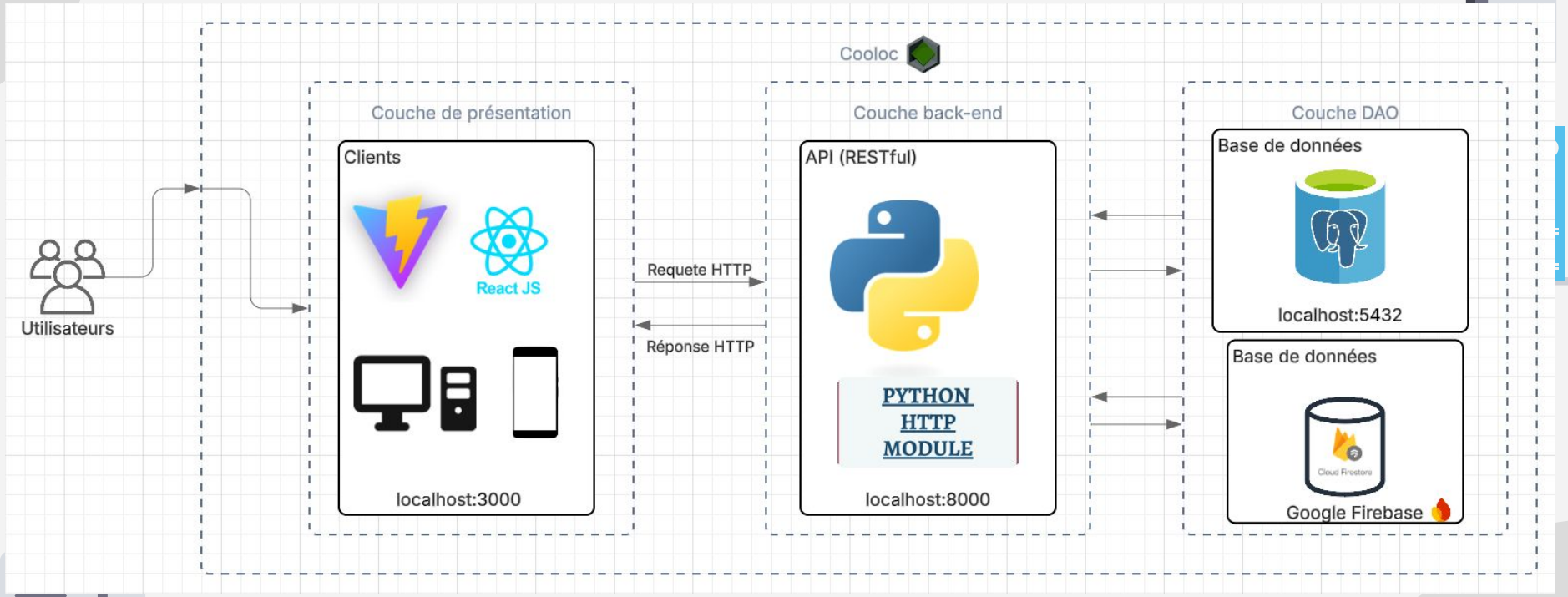








V. Conception de Cooloc - Architecture

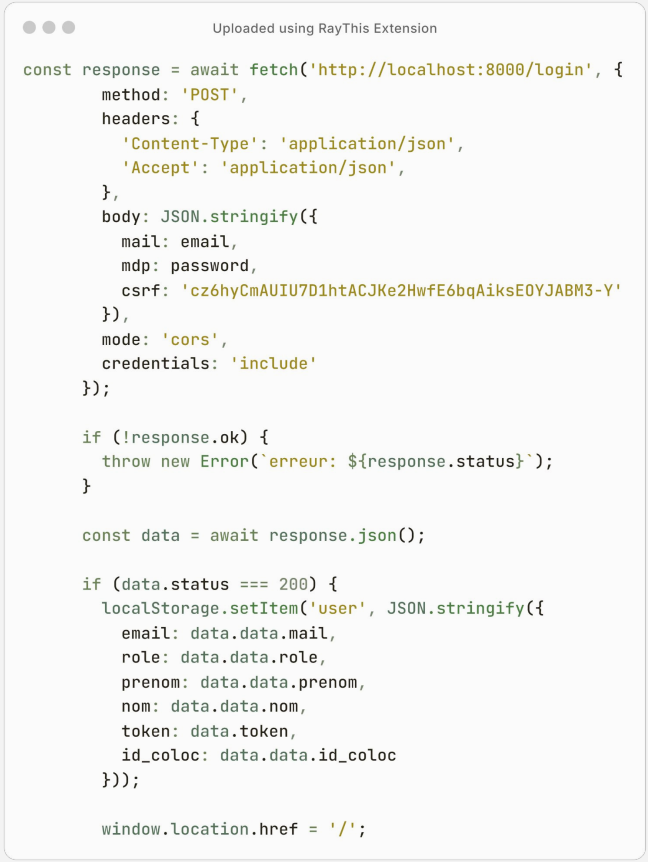


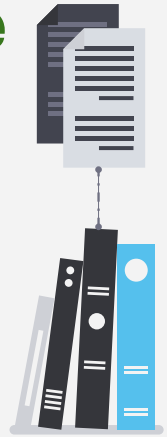
VI. Démo

VII. Développement Frontend

VII. Frontend - Navigation conditionnelle

Login.jsx





VII. Frontend - Navigation conditionnelle

App.jsx

Uploaded using RayThis Extension

```
switch (currentPage) {
      case 'home':
        return <Home />;
      case 'login':
        return <Login onRegisterClick={() => navigate('register')} />;
      case 'register':
        return <Register onLoginClick={() => navigate('login')} />;
      case 'profile':
        return <Profile />;
      case 'admin':
        return <AdminDashboard />;
      case 'a-propos':
        return <APropos />;
      case 'creation-colocation':
        return <CreationColocation />;
      case 'colocation':
        return <ColocationInfos />;
      case 'taches':
        return <Taches />;
      default:
        return <Home />;
    }
```

Header.jsx

```
<div className="header-right">
    {localUser ? (
        <div className="user-info">
            {localUser.role === 'admin' && (
                <button onClick={onAdminClick} className="btn btn-admin">
                    Tableau de Bord Admin
                </button>
            )}
            {localUser.role !== 'admin' && (
                <button onClick={onProfileClick} className="profile-button">
                    <img src="/img/icons/person.png" alt="Profile" className="profile-icon" />
                </button>
            )}
            <span className="welcome-text">Bienvenue, {localUser.prenom}</span>
            <button onClick={handleLogout} className="btn btn-logout">Déconnexion</button>
        </div>
    ) : (
        <>
            <button onClick={() => handleAuthClick(onLoginClick)} className="btn btn-login">Se connecter</button>
            <button onClick={() => handleAuthClick(onRegisterClick)} className="btn btn-register">S'inscrire</button>
        </>
    )}
</div>
```

VII. Frontend - Interfaces

AdminDashboard.jsx

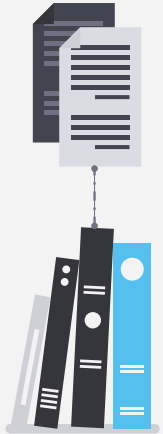
frontend > src > pages > AdminDashboard.jsx > AdminDashboard

```
function AdminDashboard() {
  <h3>Créer un Nouvel Utilisateur</h3>
  <input
    type="email"
    placeholder="Email"
    value={newUser.mail}
    onChange={(e) => setNewUser({...newUser, mail: e.target.value})}
    required
  />
  <input
    type="password"
    placeholder="Mot de passe"
    value={newUser.mdp}
    onChange={(e) => setNewUser({...newUser, mdp: e.target.value})}
    required
  />
  <input
    type="text"
    placeholder="Prénom"
    value={newUser.prenom}
    onChange={(e) => setNewUser({...newUser, prenom: e.target.value})}
    required
  />
  <input
    type="text"
    placeholder="Nom"
    value={newUser.nom}
    onChange={(e) => setNewUser({...newUser, nom: e.target.value})}
    required
  />
  <input
    type="tel"
    placeholder="Téléphone (optionnel)"
    value={newUser.num_telephone}
    onChange={(e) => setNewUser({...newUser, num_telephone: e.target.value.slice(0, 10)})}
    maxLength={10}
  />
  <select
    value={newUser.role}
    onChange={(e) => setNewUser({...newUser, role: e.target.value})}
  >
    <option value="colocataire">Colocataire</option>
    <option value="responsable">Responsable</option>
    <option value="aucun">Aucun</option>
    <option value="admin">Administrateur</option>
  </select>
  <button type="submit">Créer l'utilisateur</button>
</form>
```

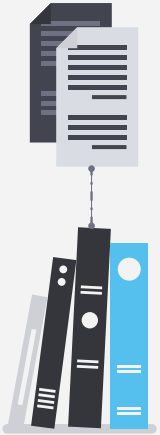
Profile.jsx

frontend > src > pages > Profile.jsx > Profile > handleSubmit > updatedUser

```
import { useState, useEffect } from 'react';
import '../styles/profile.css';

const Profile = () => {
  const [user, setUser] = useState(null);
  const [isEditing, setIsEditing] = useState(false);
  const [error, setError] = useState('');
  const [success, setSuccess] = useState('');
  const [formData, setFormData] = useState({
    email: '',
    prenom: '',
    nom: '',
    password: '',
    confirmPassword: ''
  });

  useEffect(() => {
    const storedUser = localStorage.getItem('user');
    if (storedUser) {
      const userData = JSON.parse(storedUser);
      setUser(userData);
      setFormData({
        email: userData.email,
        prenom: userData.prenom,
        nom: userData.nom,
        password: '',
        confirmPassword: ''
      });
    }
  }, []);

  const handleInputChange = (e) => {
    const { name, value } = e.target;
    setFormData(prev => ({
      ...prev,
      [name]: value
    }));
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    if (!isEditing) return;
    setError('');
    setSuccess('');

    if (formData.password !== formData.confirmPassword) {
      setError('Les mots de passe ne correspondent pas');
      return;
    }
```

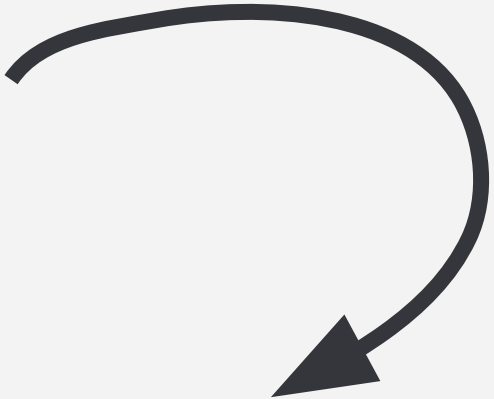


VII. Frontend - Interfaces



Footer.jsx

```jsx
import "../styles/Footer.css";

const Footer = () => {
  return (
    <footer id="footer">
      <div class="footer-top">
        <nav id="footer-nav">
          <a href="/a-propos" id="footer-link">Données personnelles</a>
          <a href="https://www.cnil.fr/fr/cookies-et-autres-traceurs/que-dit-la-loi" id="footer-link" target="_blank" rel="noopener noreferrer">Gestion des cookies</a>
          <a href="/a-propos" id="footer-link">Mentions légales &amp;amp; CGU</a>
          <a href="http://localhost:8000/docs" id="footer-link" target="_blank" rel="noopener noreferrer">Documentation technique</a>
          <a href="/a-propos" id="footer-link">Aide</a>
        </nav>
        <hr id="footer-divider" />
      </div>
      <div id="footer-bottom">
        <div id="footer-address">
          16 allée de la Marche, 92380 Garches, Hauts-De-Seine, France
        </div>
        <div id="footer-icons">
          <a href="mailto:contact@coolac.fr" id="footer-icon-link" aria-label="Mail">
            <img src="/img/icons/mail.png" alt="Mail" id="footer-icon" />
          </a>
          <a href="https://www.instagram.com/" id="footer-icon-link" target="_blank" rel="noopener noreferrer" aria-label="Instagram">
            <img src="/img/icons/insta.png" alt="Instagram" id="footer-icon" />
          </a>
          <a href="https://www.linkedin.com/" id="footer-icon-link" target="_blank" rel="noopener noreferrer" aria-label="LinkedIn">
            <img src="/img/icons/linkedin.png" alt="LinkedIn" id="footer-icon" />
          </a>
        </div>
      </div>
    </footer>
  );
};

export default Footer;
```



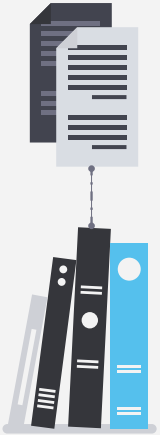
VIII. Développement Backend

VIII. Backend - Architecture & modules python

requirements.txt

Uploaded using RayThis Extension

```
psycopg2-binary
python-dotenv
pytest
pytest-cov
coverage
bcrypt
pyjwt
firebase-admin
```



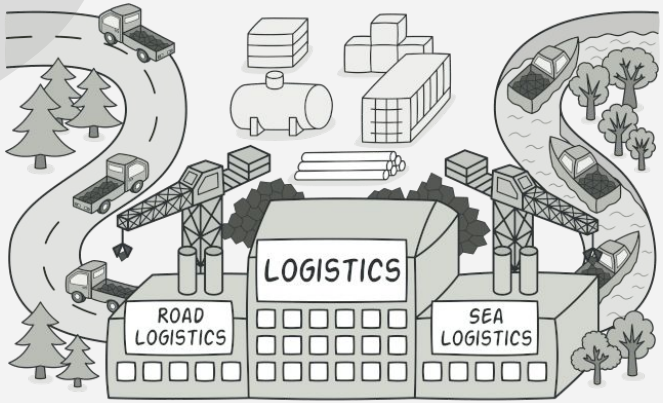
pytest

BCrypt
HASH

+

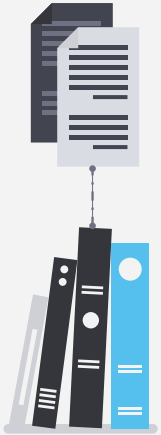
JWT

VIII. Backend - Architecture & modules python

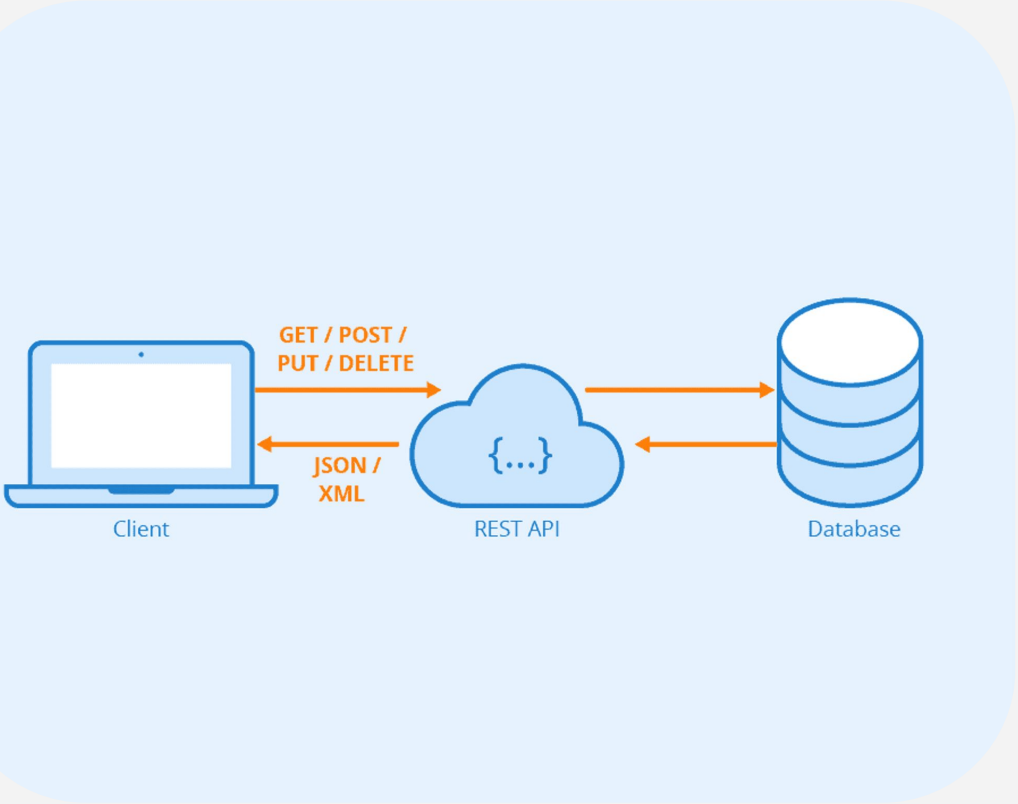


bdd/connexion.py

```python
Uploaded using RayThis Extension

class Bdd:
    """
    classe de connexion a la bdd pgsql
    """
    def __init__(self, user, mdp, nom, host, port):
        self.user = user
        self.mdp = mdp
        self.nom = nom
        self.host = host
        self.port = port

    def connexion(self):
        """
        connexion a la bdd pgsql

        :return: None
        """
        try:
            self.conn = psycopg2.connect(
                database=self.nom,
                user=self.user,
                password=self.mdp,
                host=self.host,
                port=self.port,
            )
            self.cursor = self.conn.cursor()
            print("[INFO]: Bdd OK")
        except Exception as e:
            print(f"[ERREUR]: Bdd KO : {e}")

    def close(self):
        self.conn.close()
        self.cursor.close()
        print("[INFO]: Bdd close")

class Logs:
    """
    classe de connexion a la bdd firebase
    """
    def __init__(self, cle):
        self.cle = credentials.Certificate(cle)
        self.app = firebase_admin.initialize_app(self.cle)
        self.db = firestore.client()
```



VIII. Backend - API RESTful



VIII. Backend - API RESTful

serveur.py

Uploaded using RayThis Extension

```
    def do_GET(self):
    """
    traite les req GET

    :return: None
    """
    path, parametres = self.recuperer_parametres()

    if self.path == '/':
        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()
        self.wfile.write('Serveur OK'.encode('utf-8'))

    elif self.path == '/swagger.json':
        self.send_response(200)
        self.send_header('Content-type', 'application/json')
        self.end_headers()
        self.wfile.write(json.dumps(swagger_spec).encode('utf-8'))
```

serveur.py

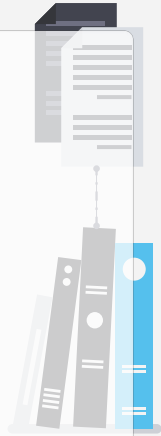
Uploaded using RayThis Extension

```
# Threading pcq sinon il galere quand ya plusieurs reqs
serveur = ThreadingHTTPServer((HOST, PORT), Serveur)
print(f'le serveur ecoute sur http://{HOST}:{PORT}')
# lance le serveur
serveur.serve_forever() # tourne en continu
serveur.server_close() # ferme le serveur si ya une action de l'utilisateur
print('stop')
```

serveur.py

Uploaded using RayThis Extension

```
class Serveur(BaseHTTPRequestHandler):
    """
    classe du serveur
    """
    # FIXME: elif ==> match case?
    def recuperer_parametres(self):
        """
        recupere les params de la requete

        :return: tuple, (path, parametres)
        """
        url = urllib.parse.urlparse(self.path)
        path = url.path

        parametres = {}
        if url.query:
            req_parametres = urllib.parse.parse_qs(url.query)

            for cle, valeur in req_parametres.items():
                parametres[cle] = valeur[0]
        return path, parametres

    def end_headers(self):
        """
        ajoute les headers CORS (accessible pour le front)

        :return: None
        """
        # front a le droit de faire des reqs
        self.send_header('Access-Control-Allow-Origin', 'http://localhost:3000')
        # les methodes autorisée
        self.send_header('Access-Control-Allow-Methods', 'GET, POST, PUT, OPTIONS')
        # les headers autorisés
        self.send_header('Access-Control-Allow-Headers', 'Content-Type, Authorization')
        # autorise les cookies
        self.send_header('Access-Control-Allow-Credentials', 'true')
        # durée d'1heure
        self.send_header('Access-Control-Max-Age', '3600')
        super().end_headers()

    def do_OPTIONS(self):
        """
        traite les req options

        :return: None
        """
        self.send_response(200)
        self.end_headers()
```



VIII. Backend - API RESTful

profil/voir_son_profile.py

Uploaded using RayThis Extension

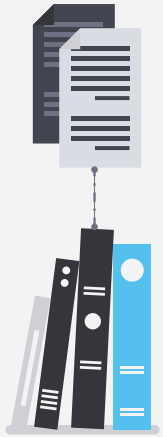
```
return {'status': 200, 'infos': utilisateurs}
```

requête postman pour voir son profil

Body Cookies Headers (8) Test Results

{} JSON Preview Visualize


```
{
    "status": 200,
    "infos": [
        {
            "id": 8,
            "mail": "test@test.com",
            "nom": "testa",
            "prenom": "testa",
            "role": "admin",
            "date_creation": "2025-05-08 00:58:14",
            "num_telephone": "0652103302",
            "id_coloc": null
        }
    ]
}
```

VIII. Backend - SQL

colocations/voir_utilisateurs.py

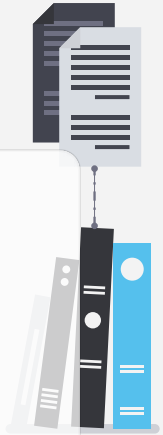
Uploaded using RayThis Extension

```python
def recup_infos(data):
    """
    recupere les informations des colocataires d'une coloc

    :param data: dict, infos envoyées par le serveur

    :return: list, infos des colocataires
    """
    coloc = data['id_colocs']
    requete = """
    SELECT u.id, u.nom, u.prenom, u.mail
    FROM Utilisateurs AS u
    JOIN Colocs AS c
    ON c.id = u.id_coloc
    WHERE c.id = %s"""
    con.cursor.execute(requete, (coloc,))
    infos = [
        {
            'id': row[0],
            'nom': row[1],
            'prenom': row[2],
            'mail': row[3]
        }
        for row in con.cursor.fetchall()
    ]
    return infos
```

colocations/taches/voir_completes.py

Uploaded using RayThis Extension

```
requete = """
SELECT t.*
FROM Taches AS t
JOIN Utilisateurs AS u
ON t.createur = u.id
JOIN Colocs AS c
ON c.id = u.id_coloc
WHERE c.id = %s
AND t.cloture = 'true'
ORDER BY t.date_crea DESC
LIMIT 4;"""
```

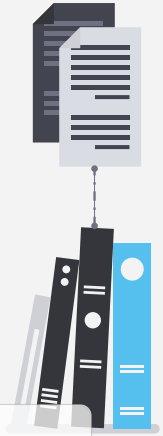


VIII. Backend - NoSQL

register.py

Uploaded using RayThis Extension

```
log = {
  'date': datetime.now(),
  'action': 'inscription',
  'mail': data['mail']
 }
logs.db.collection('Logs').add(log)
```



adm/voir_logs.py

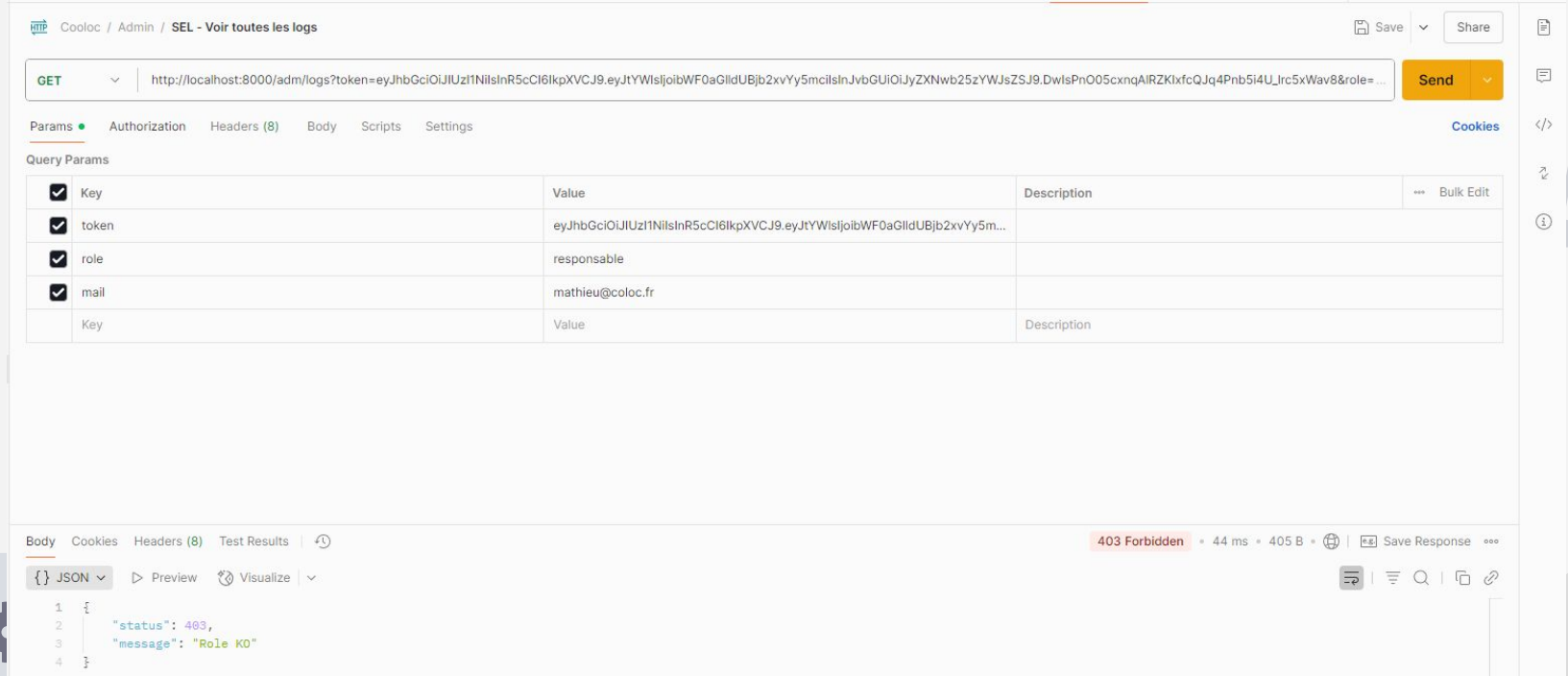
Uploaded using RayThis Extension

```
logs_bdd = logs.db.collection('Logs').stream()
# recupere TOUTEs les logs
log = []
for l in logs_bdd:
    log.append({
        'id': l.id,
        'date': l.to_dict().get('date').strftime('%Y-%m-%d %H:%M:%S'),
        # .strftime mets au bon format
        'action': l.to_dict().get('action'),
        'infos': {
            'id_coloc': l.to_dict().get('id_coloc'),
            'id_utilisateur': l.to_dict().get('id_utilisateur'),
            'id_logs': l.to_dict().get('id_logs')
        }
    })
# transforme les objets de firebase en dict
```

IX. Sécurité

IX. Sécurité - Rôles

Broken Access Control (Top 1 OWASP 2023)

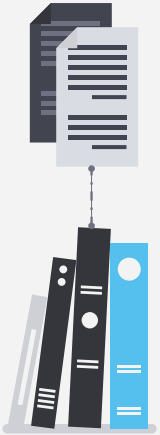


IX. Sécurité - Rôles

la majorité de mes fichiers backend

Uploaded using RayThis Extension

```
if token_decode['role'] not in ['admin']:
        return {'status': 403, 'message': 'Role KO'}
```



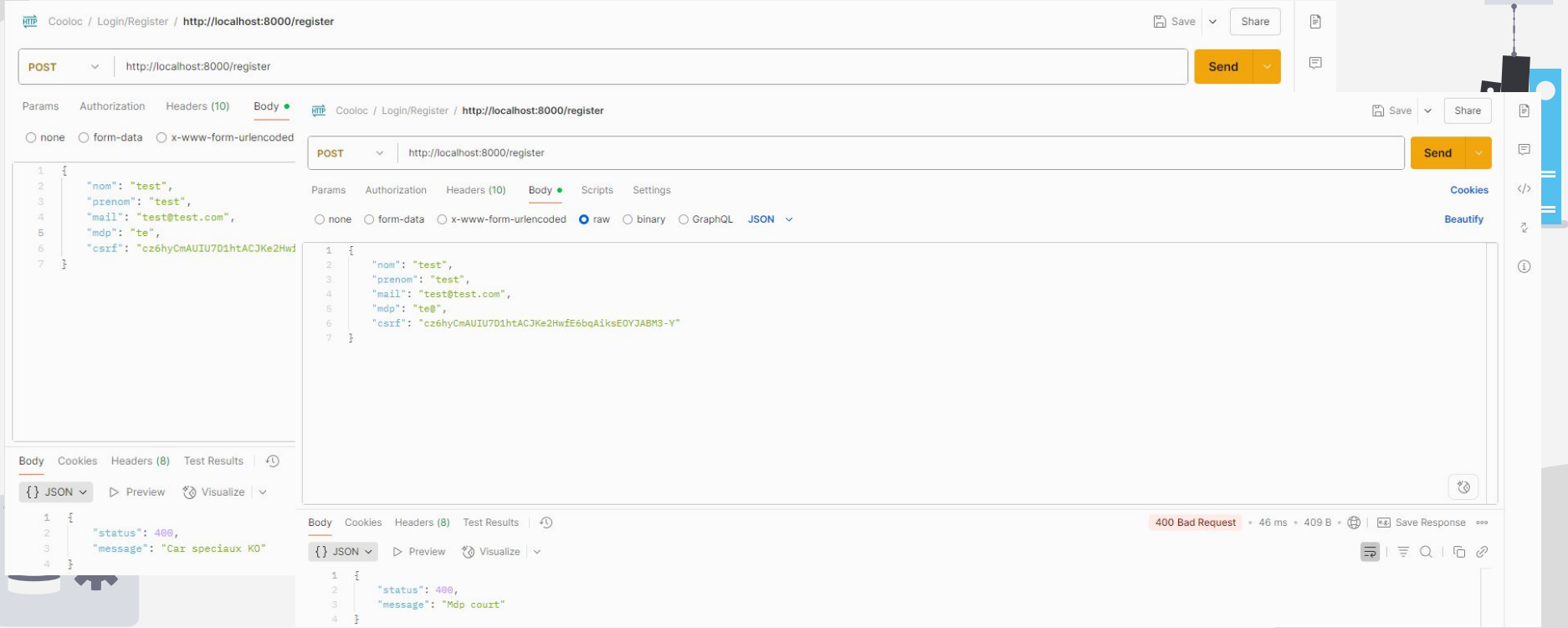
App.jsx

Uploaded using RayThis Extension

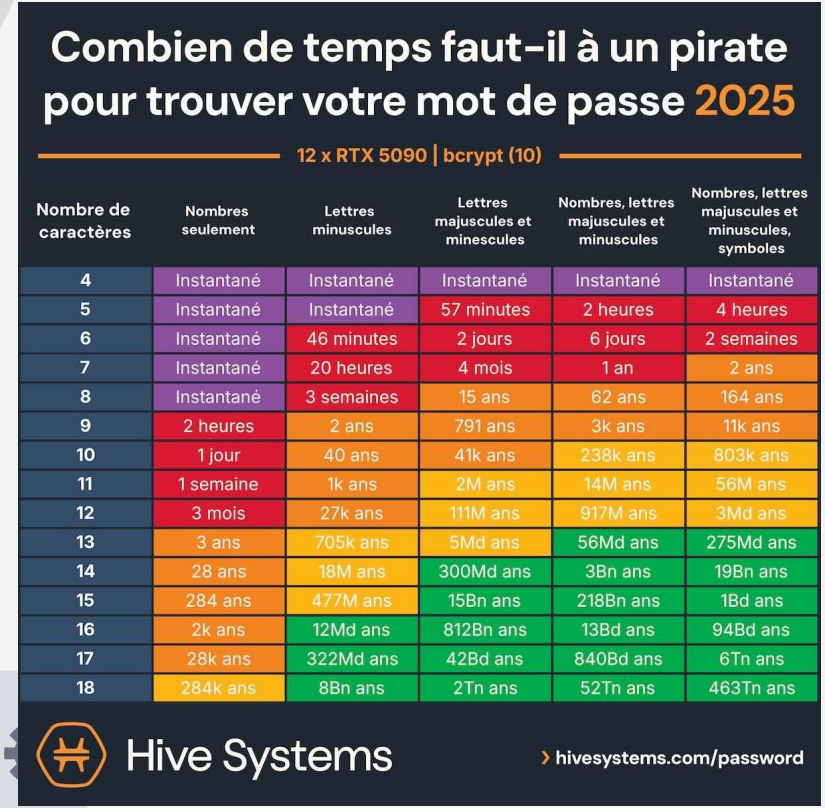
```
if (parsedUser.role === 'aucun') {
        setShowRoleModal(true);
      }

      if (parsedUser.role === 'admin' && currentPage !== 'admin') {
        window.history.pushState({}, '', '/admin');
        setCurrentPage('admin');
      }
```

IX. Sécurité - Mots de passe sécurisé

Cryptographic Failures (Top 2 OWASP 2023)



IX. Sécurité - Mots de passe sécurisé



Nombre de caractères	Nombres seulement	Lettres minuscules	Lettres majuscules et minesucles	Nombres, lettres majuscules et minuscules	Nombres, lettres majuscules et minuscules, symboles
4	Instantané	Instantané	Instantané	Instantané	Instantané
5	Instantané	Instantané	57 minutes	2 heures	4 heures
6	Instantané	46 minutes	2 jours	6 jours	2 semaines
7	Instantané	20 heures	4 mois	1 an	2 ans
8	Instantané	3 semaines	15 ans	62 ans	164 ans
9	2 heures	2 ans	791 ans	3k ans	11k ans
10	1 jour	40 ans	41k ans	238k ans	803k ans
11	1 semaine	1k ans	2M ans	14M ans	56M ans
12	3 mois	27k ans	111M ans	917M ans	3Md ans
13	3 ans	705k ans	5Md ans	56Md ans	275Md ans
14	28 ans	18M ans	300Md ans	3Bn ans	19Bn ans
15	284 ans	477M ans	15Bn ans	218Bn ans	1Bd ans
16	2k ans	12Md ans	812Bn ans	13Bd ans	94Bd ans
17	28k ans	322Md ans	42Bd ans	840Bd ans	6Tn ans
18	284k ans	8Bn ans	2Tn ans	52Tn ans	463Tn ans

register.py

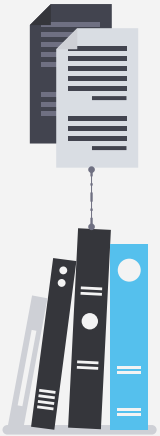
Uploaded using RayThis Extension

```python
def car_spe(mdp):
    """
    verifie que les mdp on au moins 1 caractere special

    :param mdp: str, mot de passe de l'utilisateur

    :return: bool, True si le mot de passe contient au moins un caractere special, False sinon
    """
    caracteres = "!@#$%^&*(),.?\":{}|<>"
    for c in caracteres:
        if c in mdp:
            return True
    return False
```

Uploaded using RayThis Extension

```python
if not car_spe(data['mdp']):
        return {'status': 400, 'message': 'Car speciaux KO'}

    if len(data['mdp']) < 14:
        return {'status': 400, 'message': 'Mdp court'}
```

IX. Sécurité - Mots de passe sécurisé

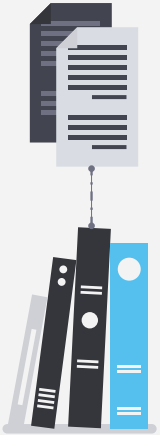


register.py

Uploaded using RayThis Extension

```python
def mdp_hash(mdp):
    """
    hashe le mot de passe de l'utilisateur avec bcrypt

    :param mdp: str, mot de passe de l'utilisateur

    :return: str, mot de passe hashe
    """
    mdp_propre = mdp.encode('utf-8')
    salt = bcrypt.gensalt(12) # 12 --> eviter bruteforce ralentir l'algo
    mdp_hashe = bcrypt.hashpw(mdp_propre, salt)
    return mdp_hashe
```

IX. Sécurité - Mots de passe sécurisé

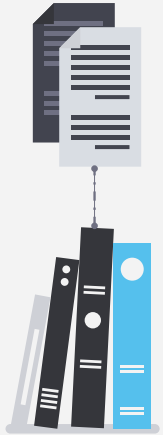


Register.jsx

Uploaded using RayThis Extension

```
if (email !== confirmEmail) {
      setError('Les adresses email ne correspondent pas');
      return;
    }

    if (password !== confirmPassword) {
      setError('Les mots de passe ne correspondent pas');
      return;
    }

    if (password.length < 14) {
      setError('Le mot de passe doit contenir au moins 14 caractères');
      return;
    }

    const specialChars = /[!@#$%^&*(),.?":{}|<>]/;
    if (!specialChars.test(password)) {
      setError('Le mot de passe doit contenir au moins un caractère spécial');
      return;
    }
```

IX. Sécurité - Mots de passe sécurisé

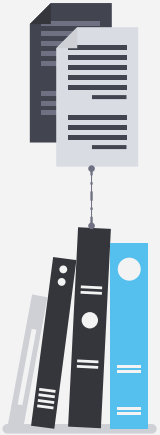


	id [PK] integer	mail character varying (50)	nom character varying (50)	prenom character varying (50)	role character varying (50)	mdp character varying (255)
1	13	mathieu@coloc.com	mathieu	coloc	colocataire	\x2432622431322456 2e4a6f484d4c716e553963353235324f4f772e772e322f3664754977303169515678593078556b51534532792e69346a6d6447…
2	11	bibou@bib.com	bibou	bibou	admin	\x2432622431322434307235765062 4e614b6c424a44734943446a52657556474e3644444473776139 4d444a6e4141303050622f37776350753 56f7…
3	16	mathieu@coloc.fr	mathieucoloc	mathieucoloc	responsable	\x2432622431322454517535664e4c6b6b4954427176745164444a4a6c4f445a3552326a6e5a4966426d5272444f6954765944465a5764364d71426…
4	17	mathieu@coloc2.fr	coloc2	coloc2	colocataire	\x2432622431322432 2f3231624170794868533070733 66d686a7463572e32794e7832784a66315a5048766c4679504257646437755692e6d6b7579…
5	10	caca@caca.com	cacio	caciooo	responsable	\x2432622431322439476f31787442334e59716448644d4a317342626e6565395241534a58576663735a44495270754573 4e685a66336859414a4a…
6	8	test@test.com	testa	testa	admin	\x2432622431322448494c697a6b2e616a34363857483461453 94e62434f556b7950676478626a712f38595a6331615836344e3751706253634b483…

IX. Sécurité - Injection XSS

Register.jsx

Injection XSS (Top 3 OWASP 2023)



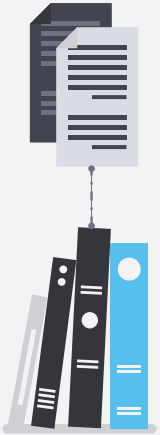
Uploaded using RayThis Extension

```
<input
                type="text"
                id="nom"
                className="form-input"
                value={nom}
                onChange={(e) => setNom(e.target.value)}
                required
                autoComplete="nom-de-famille"
              />
```

IX. Sécurité - Injection SQL

coloc/creation.py

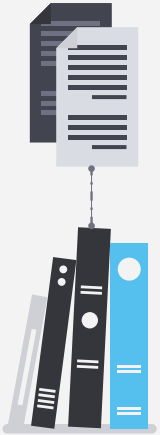
Injection SQL (Top 4 OWASP 2023)



Uploaded using RayThis Extension

```
requete = """
INSERT INTO Colocs (nom, date_crea, responsable) VALUES (%s, %s, %s) RETURNING id
"""
# RETURNING id ==> recup l'id de la coloc cree
param = (data['nom'], datetime.now(), id_utilisateur)
con.cursor.execute(requete, param)
id_coloc = con.cursor.fetchone()

if not id_coloc:
    return {'status': 500, 'message': 'Erreur interne'}

requete2 = """UPDATE Utilisateurs SET id_coloc = %s WHERE id = %s"""
con.cursor.execute(requete2, (id_coloc, id_utilisateur))
# lorsque l'user cree une coloc, il en devient responsable

log = {
  'date': datetime.now(),
  'action': 'creation coloc',
  'id_utilisateur': id_utilisateur,
  'id_coloc': id_coloc
}
logs.db.collection('Logs').add(log)

# FIXME: verifier si les actions passent avant et en fonction commit/rollback ?
con.conn.commit()
```

IX. Sécurité - Configuration



.env

Security Misconfiguration (Top 5 OWASP 2023)

Uploaded using RayThis Extension

```
BDD_NOM=Cooloc
BDD_PORT=5432
BDD_HOST=localhost
BDD_USER=postgres
BDD_MDP=postgres
JWT=a-string-secret-at-least-256-bits-long
```

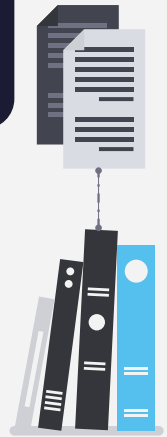
bdd/connexion.py

Uploaded using RayThis Extension

```
load_dotenv()
con = Bdd(user=os.getenv("BDD_USER"), mdp=os.getenv("BDD_MDP"), nom=os.getenv("BDD_NOM"),
host=os.getenv("BDD_HOST"), port=os.getenv("BDD_PORT"))
```

IX. Sécurité - Authentification

JWT
JSON WEB Token



Broken Authentification (Top 7 OWASP 2023)

Cooloc / Login/Register / **http://localhost:8000/login**

Save | Share

POST | http://localhost:8000/login | Send

Params | Authorization | Headers (10) | Body | Scripts | Settings | Cookies

none | form-data | x-www-form-urlencoded | raw | binary | GraphQL | JSON | Beautify

```
{
    "mail": "mathieu@coloc.fr",
    "mdp": "mathieucoloc@1234",
    "csrf": "gQS_QLRI1gzS9Fk-hT6ozC-kfwPxx-mQkOUEsJLunQQ"

}
```

Body | Cookies | Headers (8) | Test Results | 200 OK | 329 ms | 660 B | Save Response

{} JSON | Preview | Visualize

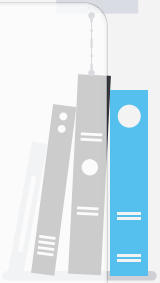
```
{
    "status": 200,
    "data": {
        "mail": "mathieu@coloc.fr",
        "role": "responsable",
        "prenom": "mathieucoloc",
        "nom": "mathieucoloc",
        "id_coloc": 27
    },
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtYWlsIjoibWF0aGlldUBjb2xvYy5mciIsInJvbGUiOiJyZXNwb25zYWJsZSJ9.DwIsPnO05cxnqAlRZKIxfcQJq4Pnb5i4U_lrc5xWav8"
}
```

IX. Sécurité - Authentification

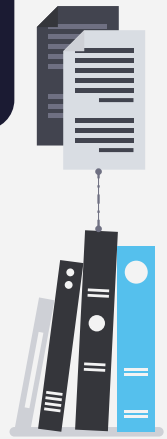
login.py

Uploaded using RayThis Extension

```python
def create_token(data):
    """
    encode avec l'ago HS256 un token jwt avec le mail & role de l'user

    :param data: dict, infos envoyées par le serveur

    :return: str, token JWT
    """
    return jwt.encode({'mail': data['mail'], 'role': data['role']}, jwt_secret,
algorithm=jwt_algo)
```



IX. Sécurité - Authentification

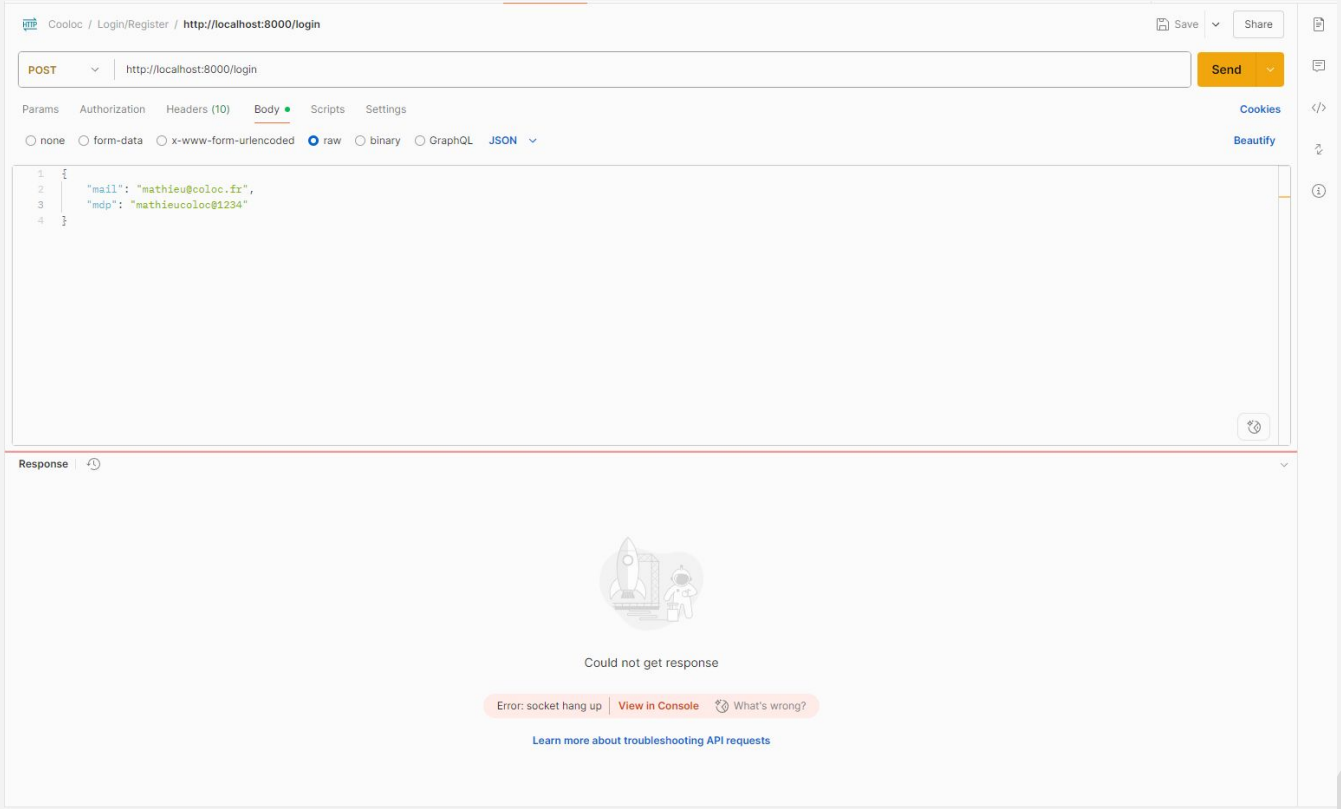


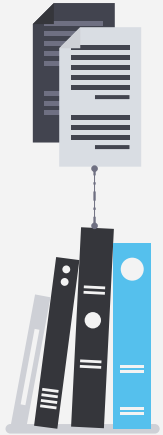
profil/voir_profil.py


```python
def verifier_token(data, token):
    """
    verifie le token JWT fourni par l'user

    :param data: dict, infos infos envoyées par le serveur
    :param token: str, token JWT de l'user

    :return: dict, status et message de la verification du token
    """
    token_decode = jwt.decode(token, jwt_secret, algorithms=[jwt_algo])

    if token_decode['mail'] != data['mail']:
        return {'status': 403, 'message': 'Token KO'}

    return {'status': 200, 'message': 'Token OK'}
```

IX. Sécurité – CSRF

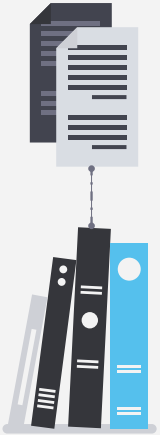
Cross Site Request Forgery (Top 8 OWASP 2023)





IX. Sécurité - CSRF

login.py

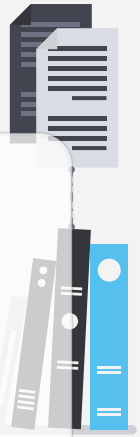


Uploaded using RayThis Extension

```python
def verifier_csrf(data):
    """
    verifie le token CSRF

    :param data: dict, infos envoyées par le serveur

    :return: dict, status et message de la verification du token CSRF
    """
    csrf = data['csrf']

    if not csrf:
        return {'status': 403, 'message': 'CSRF KO'}

    return {'status': 200, 'message': 'CSRF OK'}
```

IX. Sécurité – CSRF

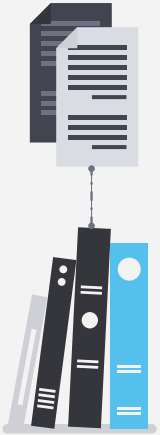
Login.jsx



Uploaded using RayThis Extension

```
body: JSON.stringify({
          mail: email,
          mdp: password,
          csrf: 'cz6hyCmAUIU7D1htACJKe2HwfE6bqAiksEOYJABM3-Y'
        }),
```

IX. Sécurité - RGPD



admin/supprimer_utilisateur.py

Uploaded using RayThis Extension

```
requete = """DELETE FROM Utilisateurs WHERE id = %s"""
    con.cursor.execute(requete, (data['id_utilisateur_supprime'],))
```

profil/supprimer.py

Uploaded using RayThis Extension

```
requete = """DELETE FROM Utilisateurs WHERE id = %s"""
    con.cursor.execute(requete, (id_utilisateur,))
```

X. DevOps

X. DevOps – Conteneurisation

backend/Dockerfile

Uploaded using RayThis Extension

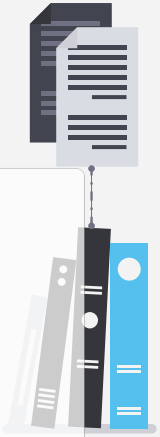
```
FROM python:3.9-slim-bookworm
WORKDIR /app

RUN pip install uv

COPY backend/requirements.txt backend/requirements.txt
RUN uv pip install --system -r backend/requirements.txt

COPY . .

EXPOSE 8000
CMD ["python", "server.py"]
```

frontend/Dockerfile

Uploaded using RayThis Extension

```
FROM node:20-alpine
WORKDIR /app

COPY frontend/package*.json ./
RUN rm -rf node_modules
RUN npm install

COPY frontend/ .
RUN npm run build

EXPOSE 3000
CMD ["npm", "run", "dev"]
```

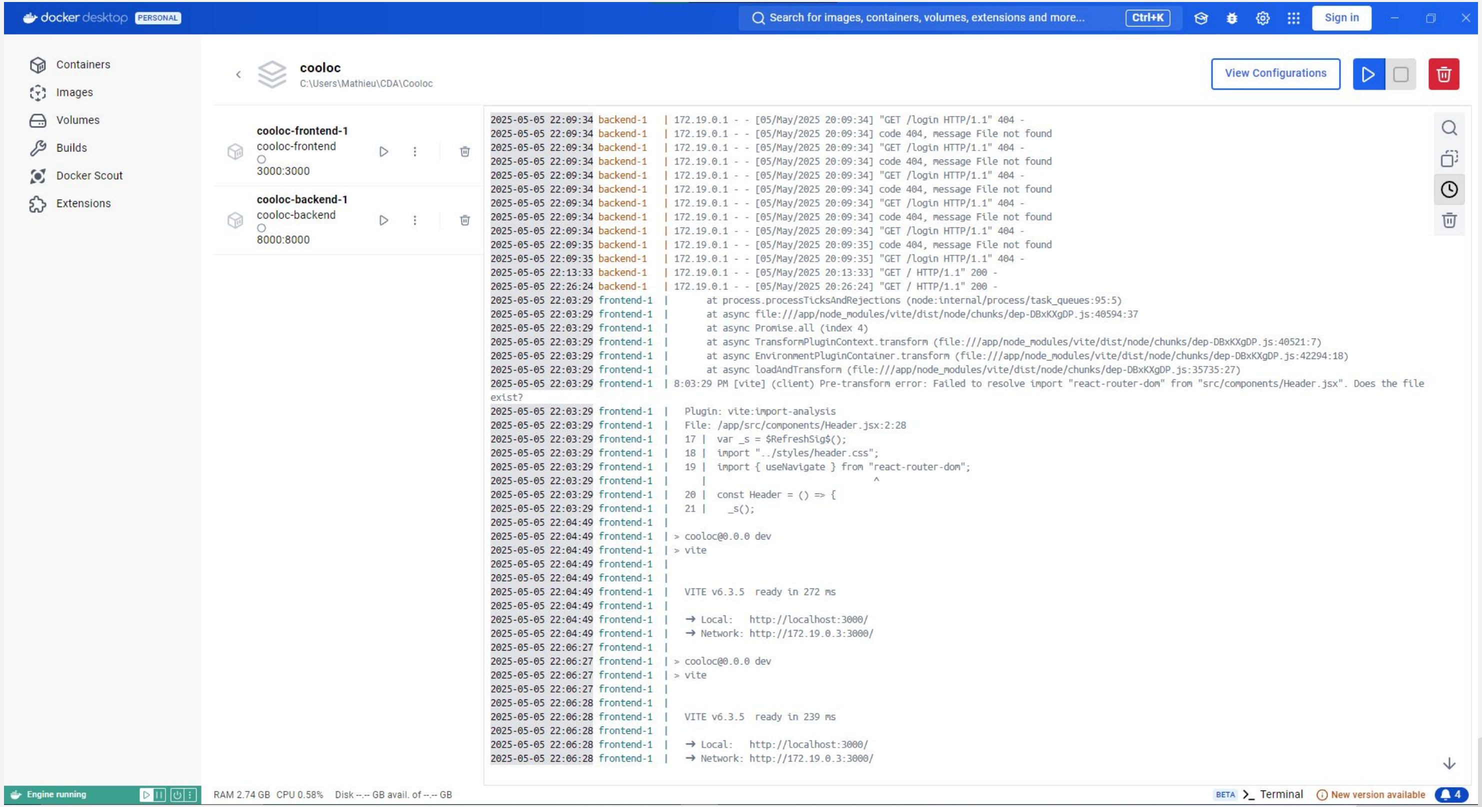
docker-compose



Uploaded using RayThis Extension

```
services:
  backend:
    build:
      context: .
      dockerfile: backend/Dockerfile
    ports:
      - "8000:8000"
    volumes:
      - ./backend:/app
    environment:
      - ENV=development

  frontend:
    build:
      context: .
      dockerfile: frontend/Dockerfile
    ports:
      - "3000:3000"
    volumes:
      - ./frontend:/app
      - /app/node_modules
    depends_on:
      - backend
```

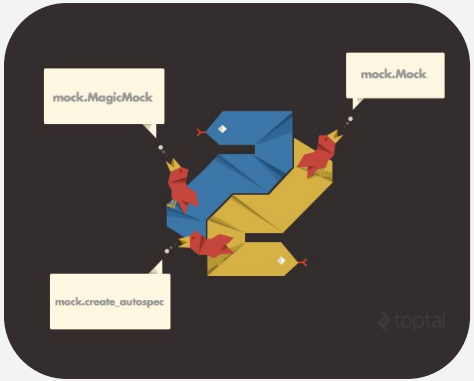
X. DevOps - Conteneurisation

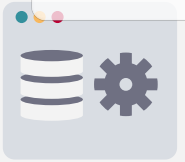


X. DevOps - Tests & validations

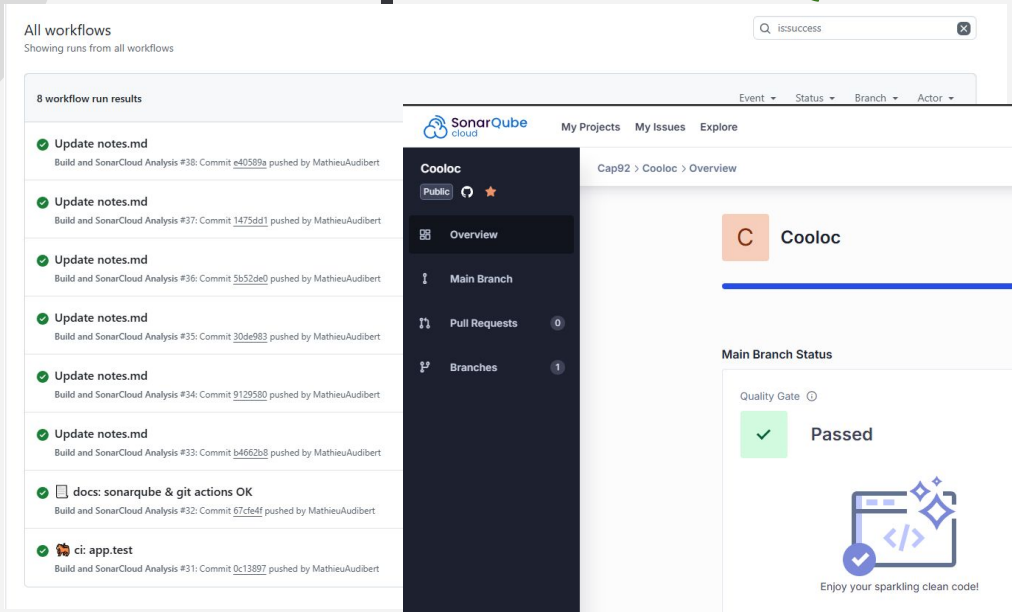
Uploaded using RayThis Extension

```python
import pytest
from unittest.mock import MagicMock, patch

@pytest.fixture(autouse=True)
def mock_firebase():
    with patch('firebase_admin.initialize_app') as mock_init, \
         patch('firebase_admin.credentials.Certificate') as mock_cert, \
         patch('firebase_admin.firestore') as mock_firestore, \
         patch('psycopg2.connect') as mock_connect:

        mock_db = MagicMock()
        mock_firestore.client.return_value = mock_db
        mock_cert.return_value = MagicMock()
        mock_connect.return_value = MagicMock()
        yield {
            'init': mock_init,
            'cert': mock_cert,
            'firestore': mock_firestore,
            'db': mock_db,
            'connect': mock_connect
        }
```



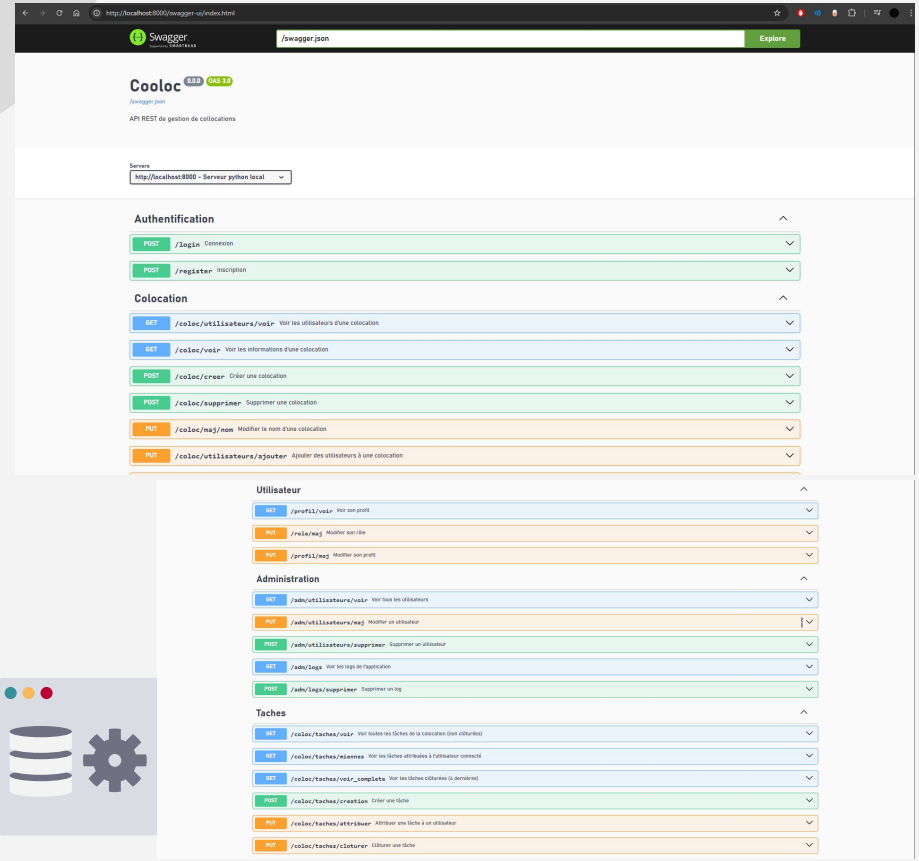


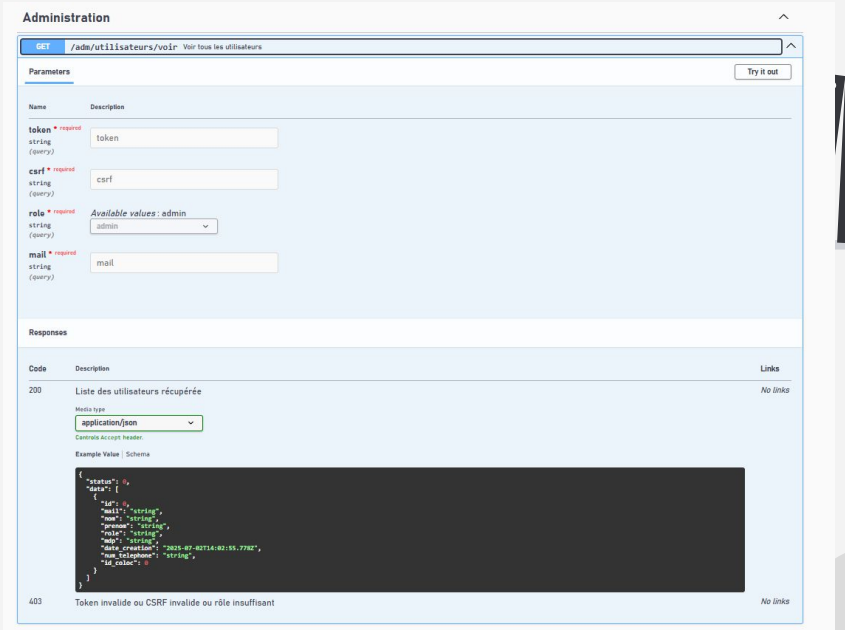
X. DevOps - SonarQube & GitHub Actions



GitHub Actions

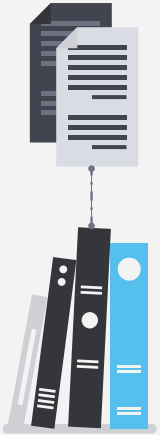
X. DevOps - Documentation technique





XI. Points d’améliorations

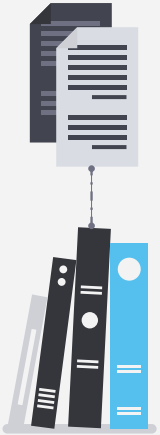
XI. Points d'améliorations - Champs



- Champs + stricts
- Contrôles sur **TOUS** les champs

XI. Points d'améliorations - Tokens

- Pas d'expiration
- Génération tokens CSRF



XII. Conclusion

Merci

Mathieu AUDIBERT - B3 Développement web & applications