

## Passage au Model Relationnel :

Etudiant(matricule, nom, prénom, groupe, codeS\*)

Section(codeS, libellé, spécialité)

Module(codeM, libelléM, coef, codeEns\*)

Examin(matricule\*, codeM\*, note)

Enseignant(codeEns, nomEns, prénomEns, grade)

Users(username, motDePass)

Failed\_attempts(ip, counter, last\_time)

## Requêtes SQL :

```
CREATE DATABASE TP_SI_BD;
```

```
CREATE TABLE Section(  
    codeS INT,  
    libellé VARCHAR(255) NOT NULL,  
    spécialité VARCHAR(255) NOT NULL,  
    PRIMARY KEY (codeS)  
);
```

```
CREATE TABLE Etudiant (  
    matricule INT,  
    nom VARCHAR(255) NOT NULL,  
    prénom VARCHAR(255) NOT NULL,  
    groupe INT NOT NULL,  
    codeS INT,  
    PRIMARY KEY (matricule),  
    FOREIGN KEY (codeS) REFERENCES Section(codeS)  
);
```

```
CREATE TABLE Enseignant(  
    codeEns INT,  
    nomEns VARCHAR(255) NOT NULL,  
    prénomEns VARCHAR(255) NOT NULL,  
    grade VARCHAR(255) NOT NULL,  
    PRIMARY KEY (codeEns)  
);
```

```
CREATE TABLE Module(  
    codeM INT,  
    libelléM VARCHAR(255) NOT NULL,  
    coef INT NOT NULL CHECK (coef > 0),  
    codeEns INT,  
    PRIMARY KEY (codeM),  
    FOREIGN KEY (codeEns) REFERENCES Enseignant(codeEns)  
);
```

```
CREATE TABLE Examin(  
    matricule INT,  
    codeM INT,  
    note DECIMAL(4,2) CHECK (note >= 0 AND note <= 20),  
    PRIMARY KEY (matricule, codeM)  
);
```

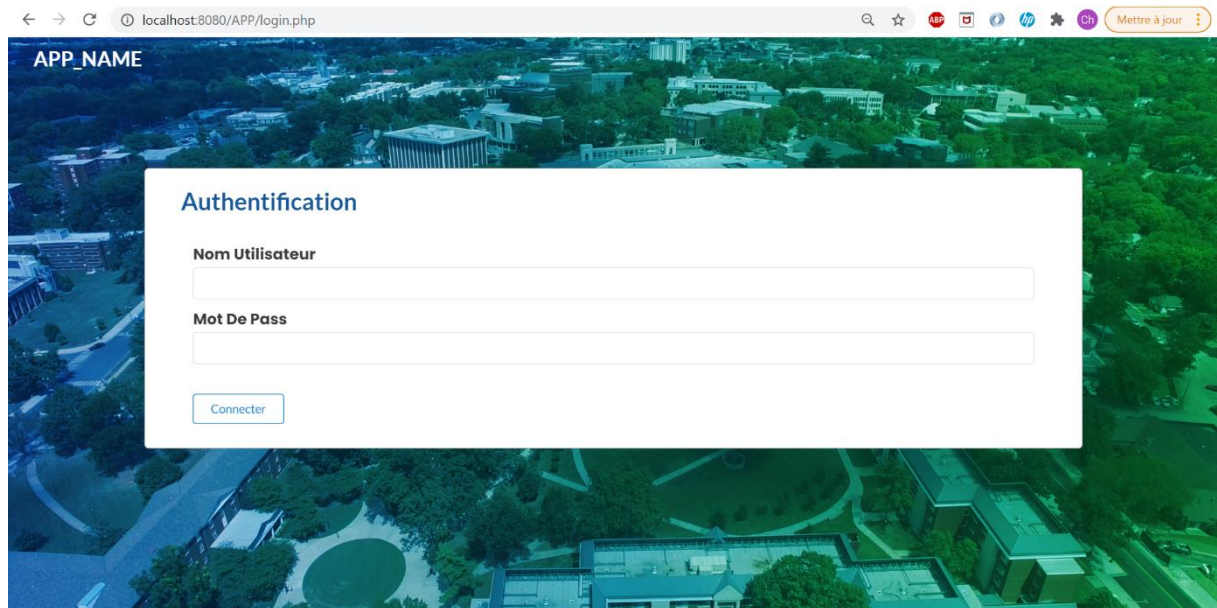
```
CREATE TABLE users(  
    username VARCHAR(255) NOT NULL,  
    motDePass VARCHAR(255) NOT NULL,  
    PRIMARY KEY (username, motDePass)  
);
```

```
CREATE TABLE failed_attempts(  
    ip VARCHAR(255),  
    counter INT DEFAULT 1 CHECK (counter <=3),  
    latest_time TIMESTAMP DEFAULT NOW() ON UPDATE NOW(),  
    PRIMARY KEY (ip)  
);
```

## Remarques concernant l'application :

- c'est une application web créé en utilisant HTML, CSS et PHP.
- les mots de passe des utilisateurs sont hachées.
- après 3 tentatives de connexion infructueuses, l'utilisateur est bloqué (en utilisant l'adresse IP). (Je sais que ce n'est pas la façon de le faire)
- tous les champs de formulaire sont protégés contre les attaques XSS (injections SQL ou malware javascript).
- toutes les entrées sont validées avant d'être enregistrées dans la base de données (exemple : la note d'examine doit être un réel entre 0 et 20).

## L'authentification:



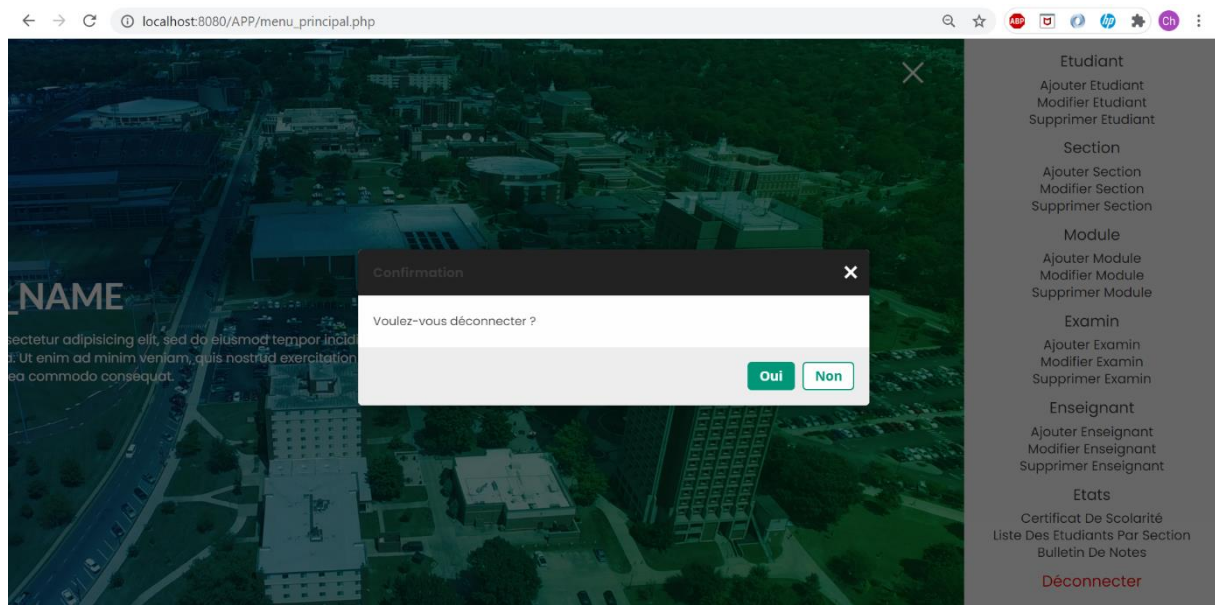
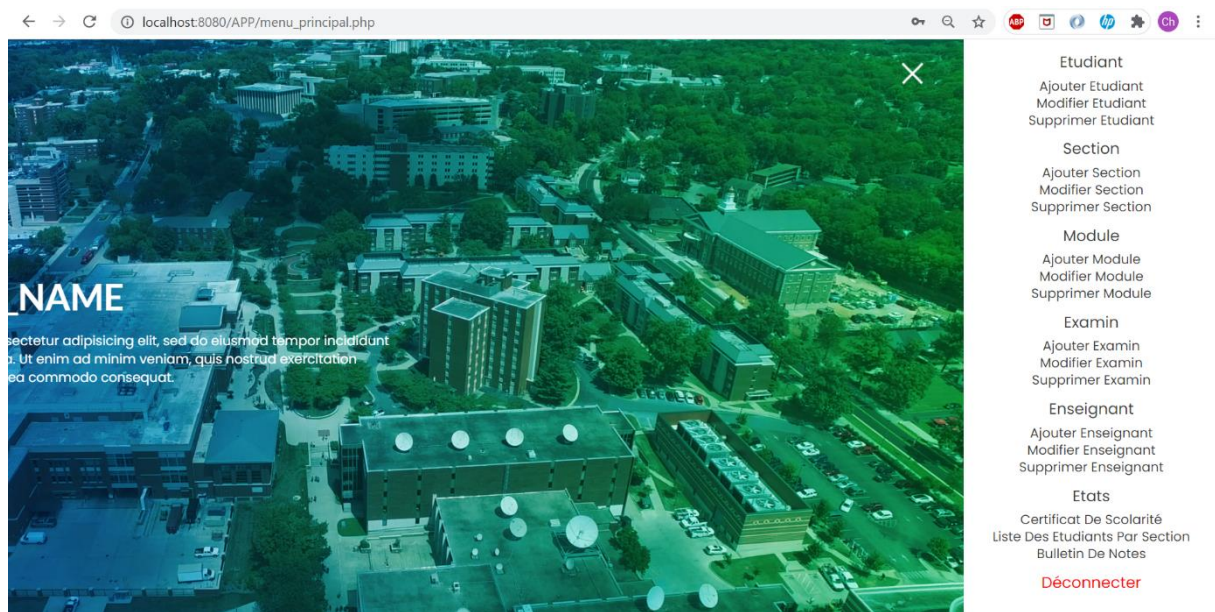
 ***Vous êtes bloqué.***

L'utilisateur ne peut pas contourner la connexion en tapant une URL vers une autre page dans la barre d'URL pour des raisons de sécurité.

Chaque utilisateur non connecté sera automatiquement redirigé vers la page de connexion lorsqu'il essaiera d'accéder à d'autres pages en utilisant la barre d'URL.

Je l'ai fait en utilisant Sessions.

J'ai également ajouté un bouton de déconnexion.

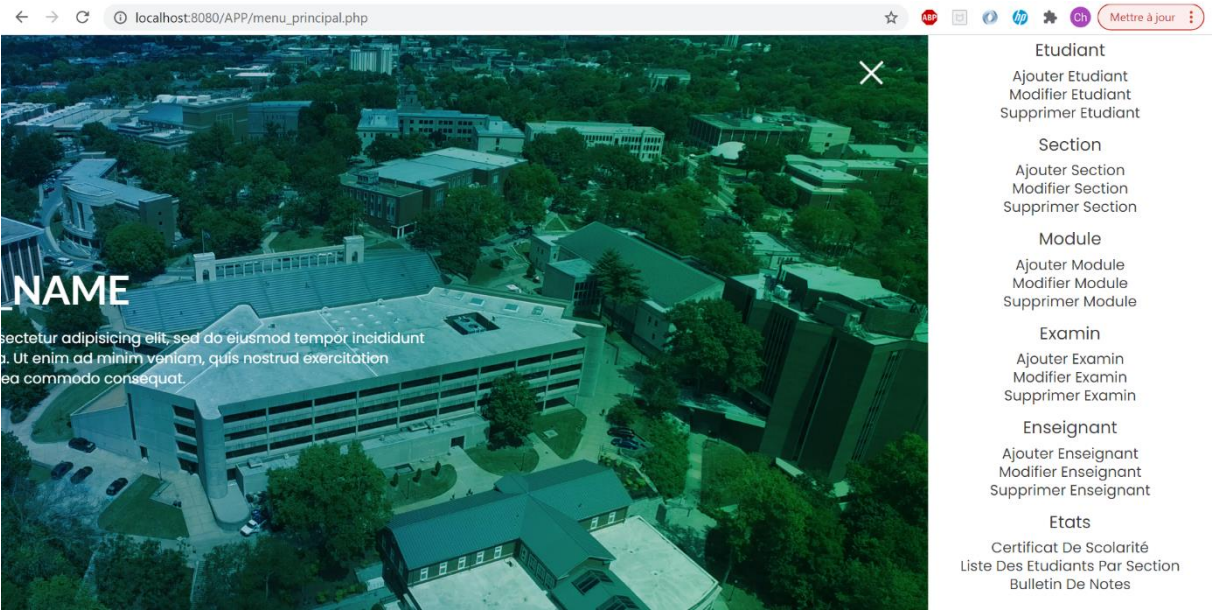
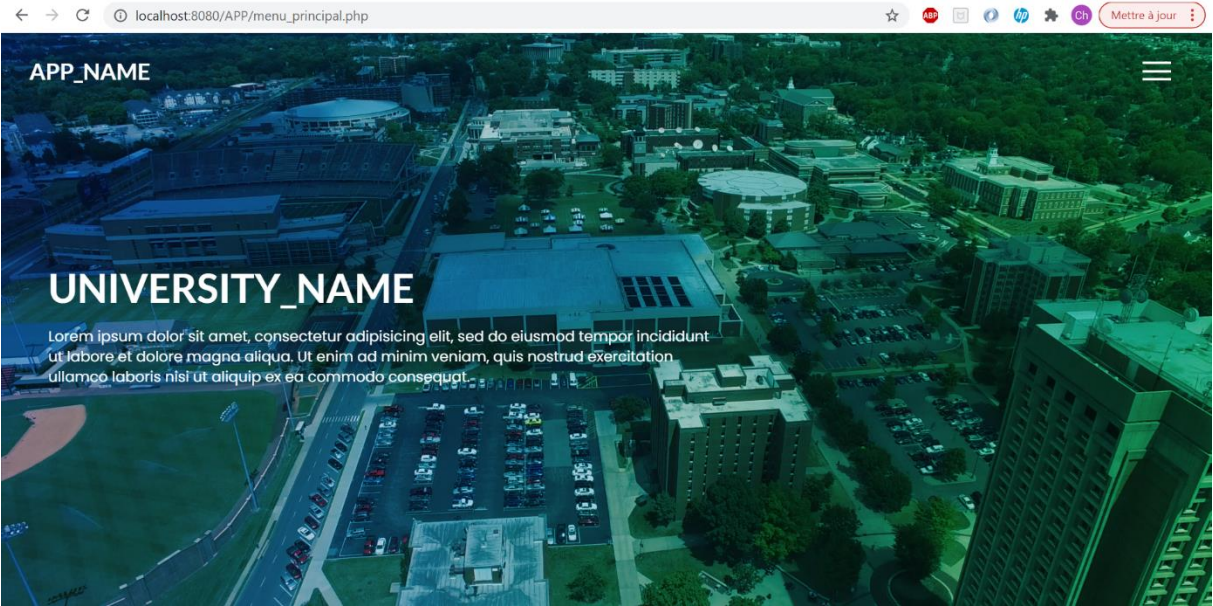


## Remarque :

Je n'ai pris que des captures d'écran des opérations (ajout, modification, suppression) sur la table « étudiant », c'est la même chose pour toutes les autres opérations sur toutes les autres tables (enseignant, section, module, examine).



# Le menu principal :



# L'ajout d'un étudiant

localhost:8080/APP/ajouter\_etudiant.php

## Ajouter Etudiant

**Matricule**  
4441 Chercher Clear

étudiant inexistant, veuillez saisir le reste des données

**Nom**  
ch\_0  
le nom doit etre une chaine de caracteres alphabetique.

**Prénom**  
prénom manquant

**codes**  
1113  
la section n'existe pas.

**Groupe**  
5  
le numero de groupe doit etre un nombre entre 1 et 4.

Ajouter Annuler

localhost:8080/APP/ajouter\_etudiant.php

APP\_NAME

## Ajouter Etudiant

**Matricule**  
4441 Chercher Clear

étudiant inexistant, veuillez saisir le reste des données

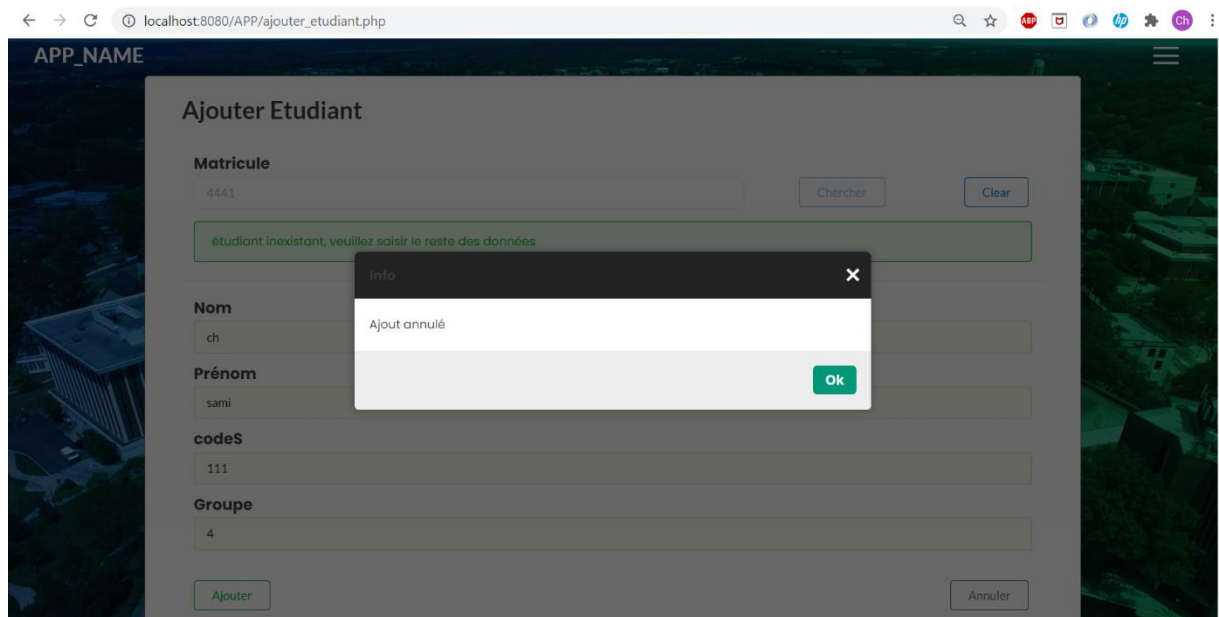
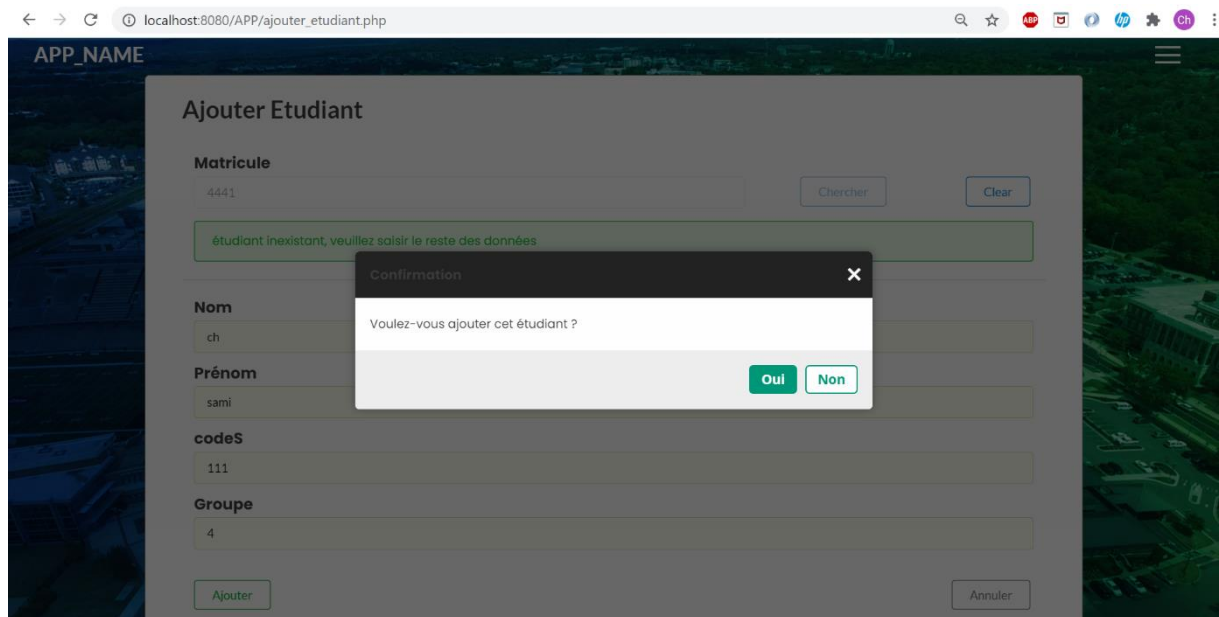
**Nom**  
ch

**Prénom**  
sami

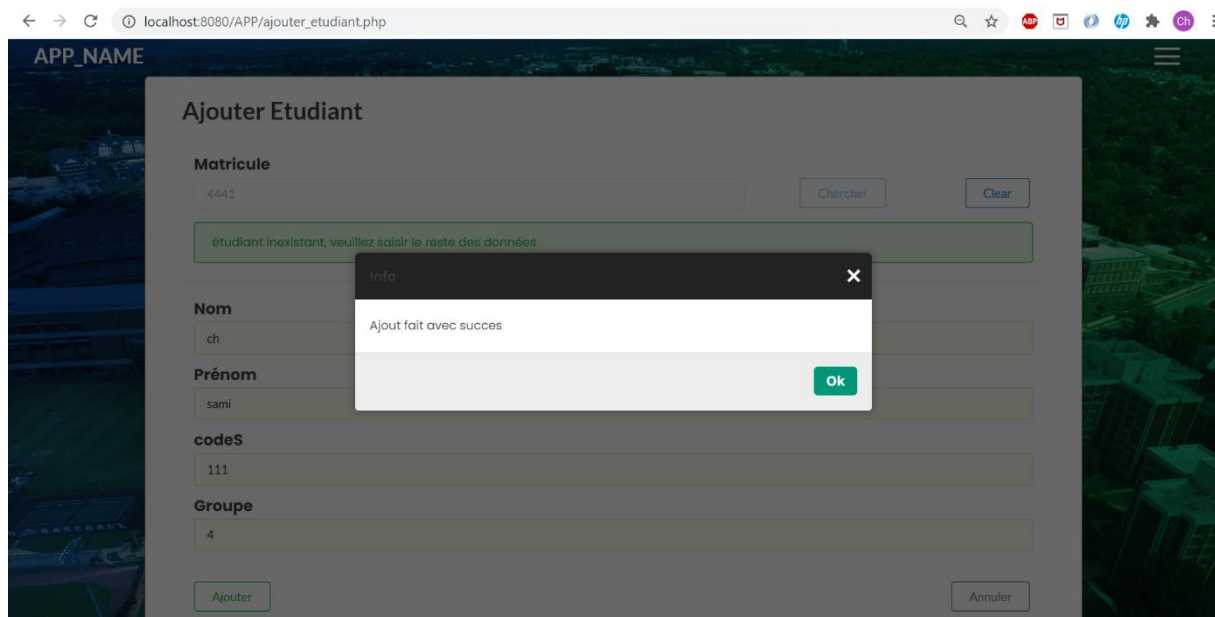
**codes**  
111

**Groupe**  
4

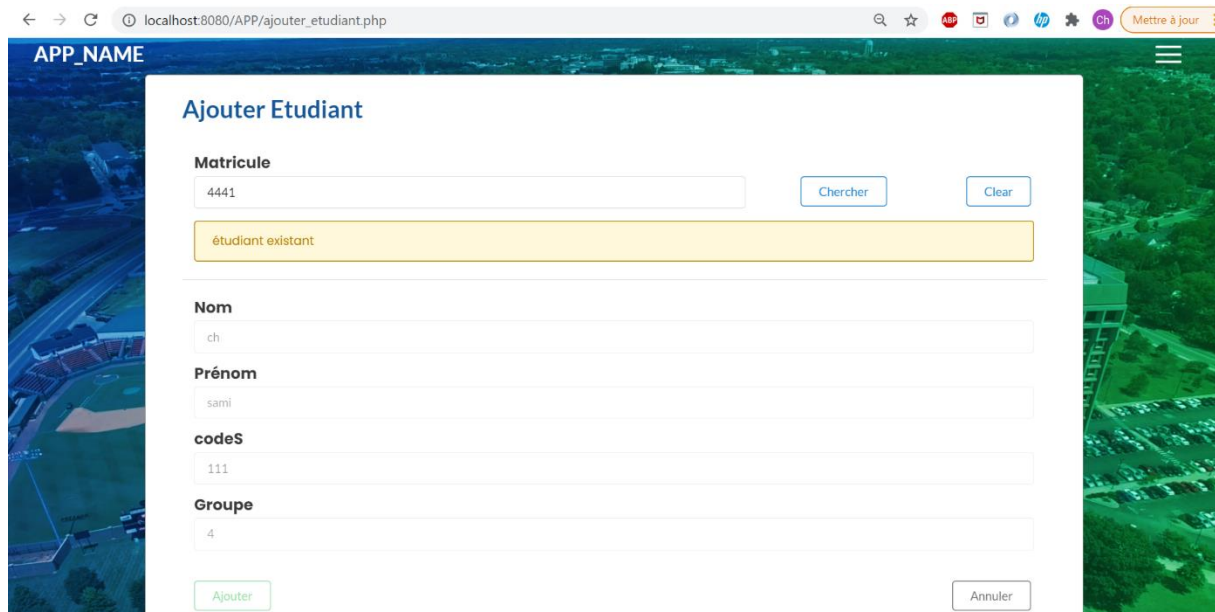
Ajouter Annuler







	matricule	nom	prénom	groupe	codeS
<input type="checkbox"/> Éditer <input type="button" value="Copier"/> <input type="button" value="Supprimer"/>	4441	ch	sami	4	111



## La modification d'un étudiant :

← → ↻ localhost:8080/APP/modifier\_etudiant.php 🔍 ☆ ABP 📧 🌐 🗄️ ch ⋮

APP\_NAME ☰

### Modifier Etudiant

**Matricule**

171731 Chercher Clear

étudiant inexistant

**Nom**

**Prénom**

**codeS**

**Groupe**

Modifier Annuler

← → ↻ localhost:8080/APP/modifier\_etudiant.php 🔍 ☆ ABP 📧 🌐 🗄️ ch ⋮

APP\_NAME ☰

### Modifier Etudiant

**Matricule**

4441 Chercher Clear

étudiant existant, veuillez modifier les données

**Nom**

ch

**Prénom**

sami

**codeS**

111

**Groupe**

4

Modifier Annuler

**Nom**

8\_ch

le nom doit etre une chaine de caracteres alphabetique.

**Prénom**

prénom manquant

**codeS**

112

la section n'existe pas.

**Groupe**

5

le numero de groupe doit etre un nombre entre 1 et 4.

**Nom**

chanane

**Prénom**

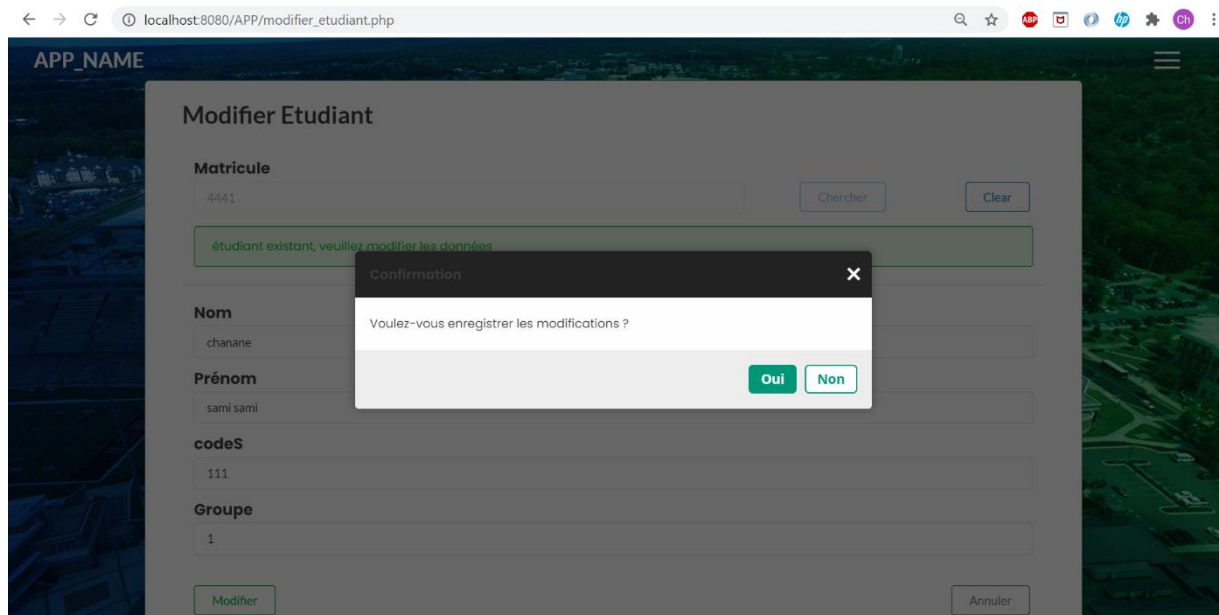
sami sami

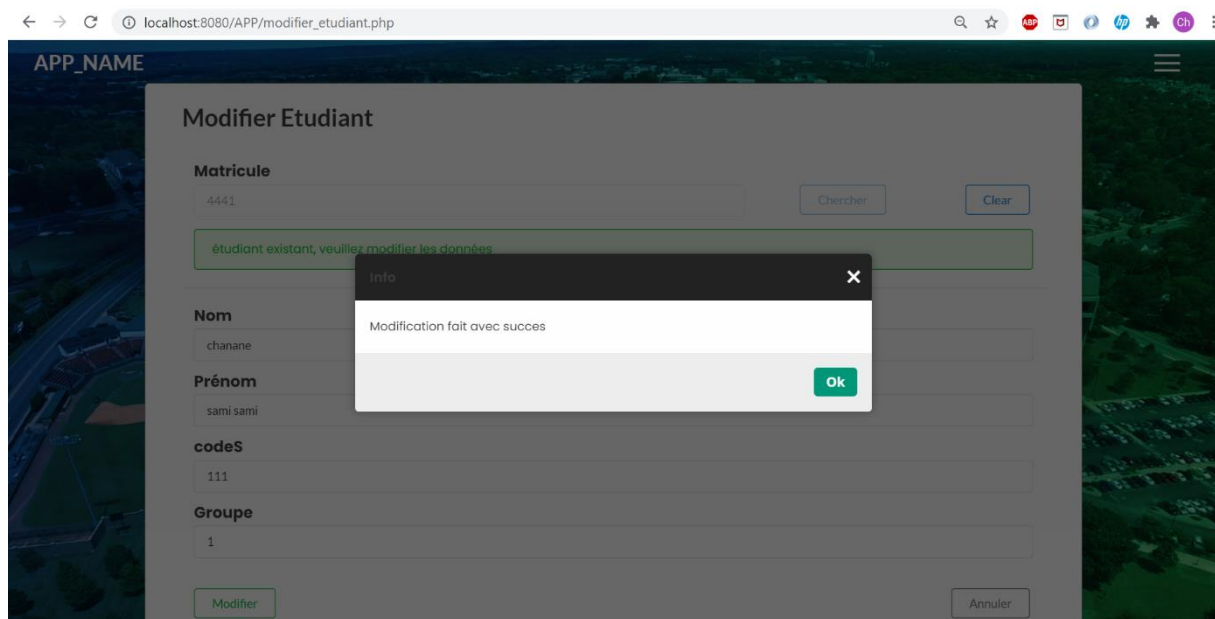
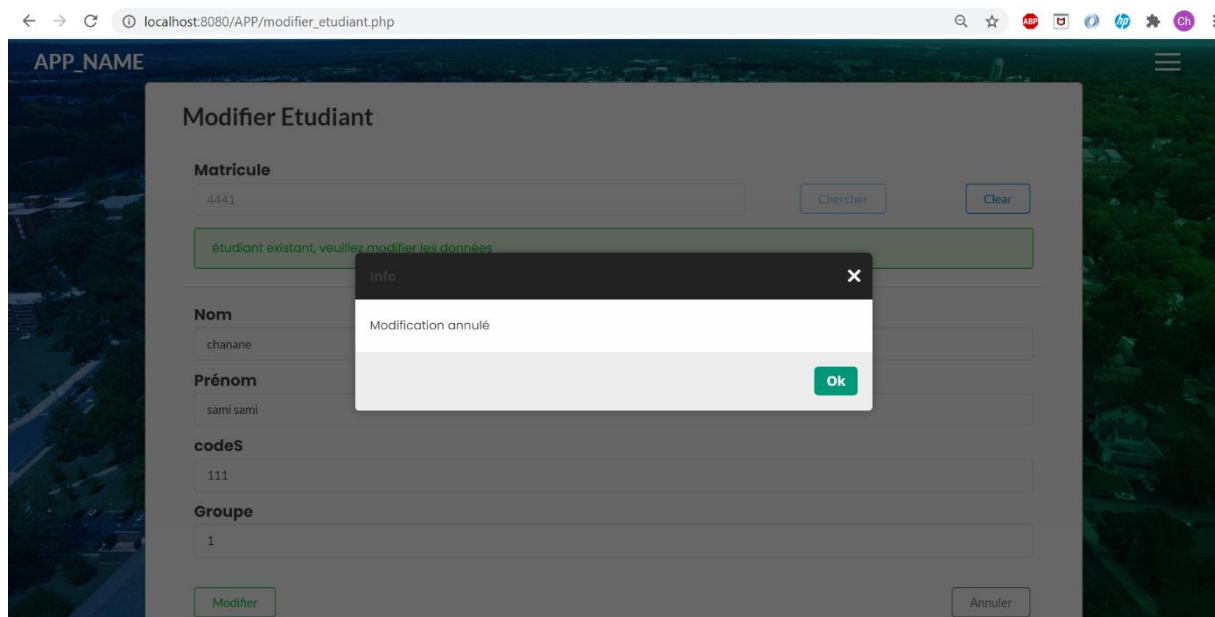
**codeS**



111

**Groupe**

1





	matricule	nom	prénom	groupe	codeS
<input type="checkbox"/>  Éditer  Copier  Supprimer	4441	chanane	sami sami	1	111



# La suppression d'un étudiant :

← → ↻ localhost:8080/APP/supprimer\_etudiant.php 🔍 ☆ APP 📧 🌐 ⚙️ Ch ⋮

APP\_NAME

### Supprimer Etudiant

**Matricule**

444777

Chercher Clear

étudiant inexistant.

**Nom**

**Prénom**

**codeS**

**Groupe**

Supprimer Annuler

← → ↻ localhost:8080/APP/supprimer\_etudiant.php 🔍 ☆ APP 📧 🌐 ⚙️ Ch ⋮

APP\_NAME

### Supprimer Etudiant

**Matricule**

4441

Chercher Clear

étudiant existant.

**Nom**

chanane

**Prénom**

sami sami

**codeS**

111

**Groupe**

1

Supprimer Annuler

