



RAPPORT DEVELOPPEMENT D'APPLICATIONS MOBILE

Projet : Application mobile de gestion des étudiants

Enseignant : M. GANSBEOGO

I. INTRODUCTION

Dans un monde où la technologie est en plein essor, il y a la présence d'outils sophistiqués nous permettant de gérer facilement nos besoins, d'acquérir plus de temps sur la gestion.

C'est dans ce contexte que nous allons aborder notre projet qui permettra la gestion des étudiants dans une école.

II. OBJECTIF

Notre objectif en réalisant ce projet, c'est de faciliter la gestion (basique) des étudiants d'une école.

Ainsi, c'est de :

- Développer une application mobile simple
- Implémenter un CRUD complet
- Connecter une application mobile à une API REST
- Utiliser une base de données relationnelle (MySQL)

III. TECHNOLOGIES UTILISÉES

Couche	Technologie
Mobile	React Native
Backend	Node.js, Express
Base de données	MySQL
Échange	JSON
Outils et autres	VS Code, Rest-Client, MySQL-Workbench

IV.ANALYSE ET CONCEPTION

En analysant le projet et en vue de sa conception, et comme besoins fonctionnels misent en place, nous avons réalisés à pile aux fonctionnalités demandées.

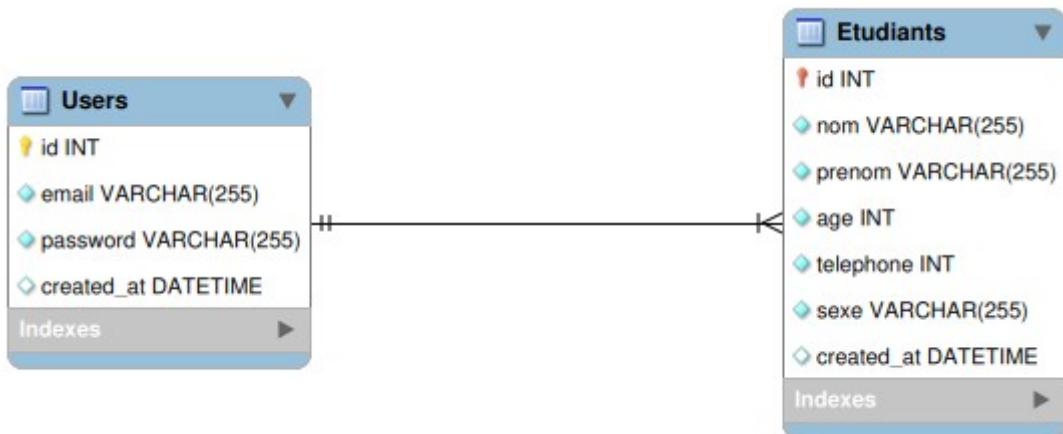
Sur lesquelles fonctionnalités, nous avons :

- Ajouter un étudiant
- Modifier un étudiant
- Supprimer un étudiant
- Consulter la liste des étudiants
- Voir le détail d'un étudiant

V. MODÉLISATION DES DONNÉES

Pour la modélisation, nous avons utilisé MySQL-Workbench.
Ci-dessous,

- Tables : **Users** et **Etudiants**



- Scripts SQL de la base de données

Création de la base de données :

```
CREATE DATABASE gestion_etudiants
DEFAULT CHARACTER_SET utf8mb4
DEFAULT COLLATE utf8mb4_unicode_ci ;
```

Création de la table **Users** :

```
CREATE TABLE users (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(255) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Création de la table Etudiants

```
CREATE TABLE etudiants (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(255) NOT NULL,
    prenom VARCHAR(255) NOT NULL,
    age INT NOT NULL,
    telephone INT NOT NULL UNIQUE,
    sexe VARCHAR(255) NOT NULL,
    userId INT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT fk_users FOREIGN KEY (userId) REFERENCES
    users(id) ON DELETE CASCADE
);
```

VI. RÉALISATION TECHNIQUE

Backend (Node.js / Express)		Frontend (React Native)
Création de l'API REST		Affichage de la liste des étudiants
Routes CRUD	GET /liste/:email	Recherche d'étudiant
	POST /ajouter	Affichage de détails
	PUT /modifier/:id	Affichage de statistique
	DELETE /supprimer/:id	Formulaire d'ajout et de modification Formulaire d'authentification (connexion et inscription)
Connexion à MySQL		Navigation entre écrans (5 écrans)
		Appels API avec axios
		Gestions des états (useState, useEffect)

VII. TETS ET RÉSULTATS

Test réalisés au cours de la conception du projet :

- Test routes API avec REST-CLIENT
- Vérification des opérations CRUD

Résultats Obtenu :

- Application fonctionnelle à notre niveau
- Données persistées dans MySQL
- Synchronisation frontend – backend réussie

VIII. CONCLUSION ET PERSPECTIVES

En réalisant ce projet, nous avons appris à gérer les données, et à réalisé une communication fluide entre un backend et un frontend.

Les difficultés rencontrés au cours de ce projet, c'était surtout la connexion API – Mobile, la gestion des erreurs, et la persistance des données côté frontend.

Ci-joint les écrans de mon application

