

المدرسة العليا لأساتذة التعليم التقني المحمدية جامعة الحسن الثاني بالدار البيضاء

Examen final

FI BDCC S3 : Bases de données semi structurées

EM2 - SGBD NoSQL

Mohamed Hammane BDCC II

- I. Création de l'application Backend :
- 1. Schéma User:

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;
const userSchema = new Schema({
 username: {
   type: String,
   required: true,
   unique: true,
   trim: true,
   minlength: 3
 gender : { type : String },
 dob:{type: Date},
 news:{type : Boolean},
  email : {type : String, required: true},
 photo : {type : String,required: true}
}, {
  timestamps: true,
});
const User = mongoose.model('User', userSchema);
module.exports = User;
```

2. Les opérations de CRUD sur l'entité user :

```
//Retrouver l'utilisateur ayant l'_id « id »(work)
router.route('/:id').get((req, res) => {
   User.findById(req.params.id)
        .then(user => res.json(user))
        .catch(err => res.status(400).json('Error: ' + err));
});
```

Pour manipuler les différentes opérations de la sélection j'ai créé :







المدرسة العليا لأساتذة التعليم التقني المحمدية جامعة الحسن الثاني بالدار البيضاء

```
//Retrouve <size> utilisateur à partir de l'offset <page> x <size></size> (work)
//users/:page/:size?search=toto&gender=1&dob=-1
router.route('/:page/:size').get((req, res) => {
 User.countDocuments({},function(err,count)
    let search = (req.query.search === undefined)?'':req.query.search ;
    let gender = (req.query.gender === undefined)?0:parseInt(req.query.gender);
    let dob = (req.query.dob === undefined)?0:parseInt(req.query.dob) ;
    let sort = {createdAt:'asc'};
    if(gender !== 0) sort = {gender:gender};
    else if(dob !== 0) sort = {dob:dob};
    let q = {};
    if(search !== '') q = {"username": { "$regex": search, "$options": "i" }};
    User.find(q).skip(parseInt(req.params.size) * (Math.max(1, parseInt(req.params.page))-1)).limit(parseInt(req.params.size))
      .sort(sort)
      .then(
        function(user){
         let s = count;
if(search !== '') s = user.length;
          return res.json( {
            users:user,
            infos:{
              page:parseInt(req.params.page),
              per_page:parseInt(req.params.size),
               sup:{
                search : req.query.search ,
                gender : req.query.gender ,
                dob : req.query.dob,
                sort:sort
      .catch(err => res.status(400).json('Error: ' + err));
 3)
```





المدرسة العليا لأساتذة التعليم التقني المحمدية جامعة الحسن الثاني بالدار البيضاء

```
//Créer un nouvel utilisateur (work)
router.route('/').post((req, res) => {
  const username = req.body.username;
  const gender = req.body.gender;
  const dob = req.body.dob;
  const news = req.body.news;
  const email = req.body.email;
  const photo = req.body.photo;
  const newUser = new User(
      username, gender, dob, news, email, photo
  newUser.save()
    .then(() => res.json('User added!'))
    .catch(err => res.status(400).json('Error: ' + err));
});
router.route('/:id').put((req, res) => {
  User.findById(req.params.id)
    .then(user => {
      user.username = req.body.username;
     user.gender = req.body.gender;
      user.dob = Date.parse(req.body.dob);
      user.news = req.body.news;
      user.email = req.body.email;
     user.photo = req.body.photo;
      user.save()
        .then(() => res.json('User updated!'))
        .catch(err => res.status(400).json('Error: ' + err));
    .catch(err => res.status(400).json('Error: ' + err));
});
//Supprime l'utilisateur ayant l'_id « id » (work)
router.route('/:id').delete((req, res) => {
  User.findByIdAndDelete(req.params.id)
    .then(() => res.json('User deleted.'))
    .catch(err => res.status(400).json('Error: ' + err));
```

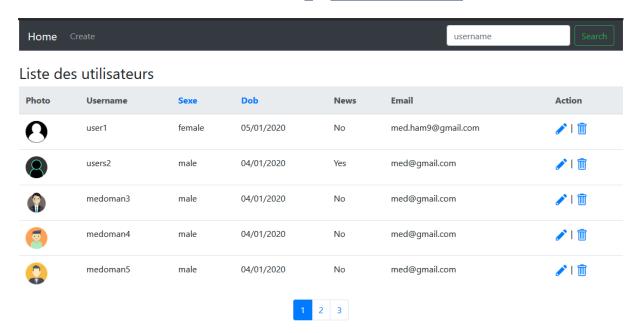
- II. Création de l'application FrontEnd :
- 1. Page d'accueil:

UNIVERSITÉ HASSAN II DE CASABLANCA

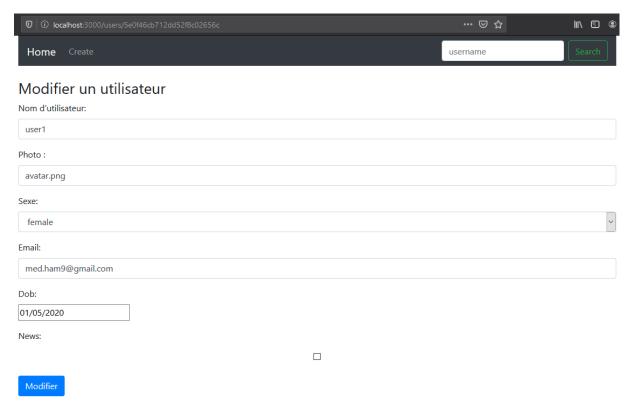




المدرسة العليا لأساتذة التعليم التقني المحمدية جامعة الحسن الثاني بالدار البيضاء



Par default on affiche 5 utilisateurs par page, cette inclure la suppression et modification d'un utilisateur.



2. Création d'un utilisateur :

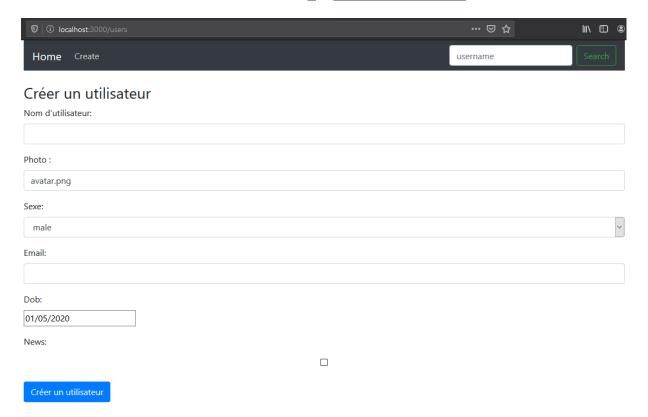
ECOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT TECHNIQUE DE MOHAMMEDIA

UNIVERSITÉ HASSAN II DE CASABLANCA

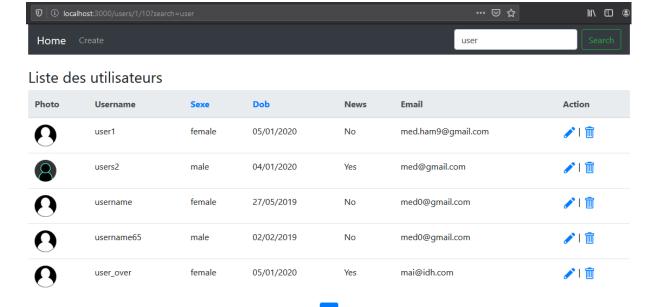




المدرسة العليا لأساتذة التعليم التقني المحمدية جامعة الحسن الثاني بالدار البيضاء



3. Recherche par username:



4. Le tri par dob ou sexe:

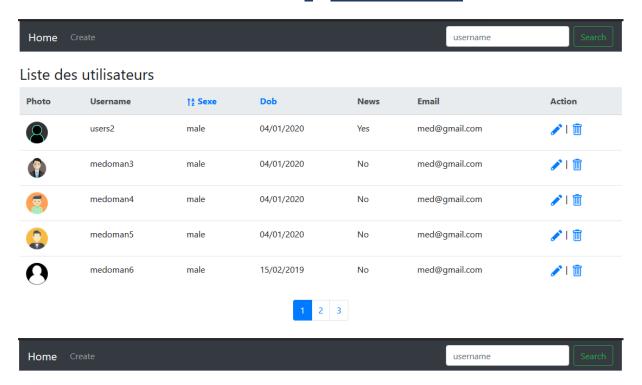
ECOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT TECHNIQUE DE MOHAMMEDIA

UNIVERSITÉ HASSAN II DE CASABLANCA





المدرسة العليا لأساتذة التعليم التقني المحمدية جامعة الحسن الثاني بالدار البيضاء



Liste des utilisateurs

Photo	Username	↓ª Sexe	Dob	News	Email	Action
0	user1	female	05/01/2020	No	med.ham9@gmail.com	∤ 🛅
0	username	female	27/05/2019	No	med0@gmail.com	∤ 🛅
0	user_over	female	05/01/2020	Yes	mai@idh.com	∦ I 🛅
	test1	female	05/01/2015	No	test@gmail.com	∦ l 🛅
8	users2	male	04/01/2020	Yes	med@gmail.com	∥



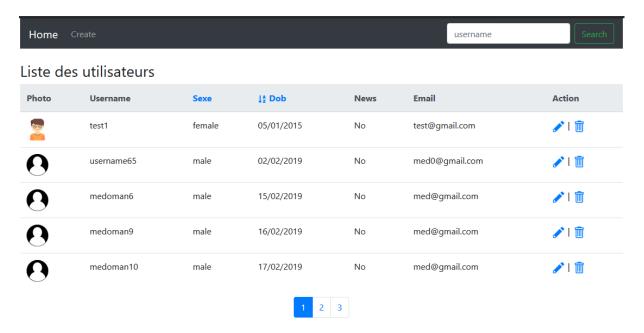
ECOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT TECHNIQUE DE MOHAMMEDIA

UNIVERSITÉ HASSAN II DE CASABLANCA





المدرسة العليا لأساتذة التعليم التقني المحمدية جامعة الحسن الثاني بالدار البيضاء



Si vous cliquez sur sexe ou dob le site sera trier par ce dernier.

Et voici le projet en GitHub : https://github.com/Medomane/Examen