

## 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 :

```
//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),
              size:s,
              sup:{
                search : req.query.search ,
                gender : req.query.gender ,
                dob : req.query.dob,
                sort:sort
              }
            }
          })
        }
      )
    )
  ).catch(err => res.status(400).json('Error: ' + err));
});
```

```
//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));
});

//Met à jour l'utilisateur ayant l'id « id » (work)
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 :

Home Create

username

Search

## Liste des utilisateurs

Photo	Username	Sexe	Dob	News	Email	Action
	user1	female	05/01/2020	No	med.ham9@gmail.com	   
	users2	male	04/01/2020	Yes	med@gmail.com	   
	medoman3	male	04/01/2020	No	med@gmail.com	   
	medoman4	male	04/01/2020	No	med@gmail.com	   
	medoman5	male	04/01/2020	No	med@gmail.com	   

1 2 3

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

localhost:3000/users/5e0f46cb712dd52f8c02656c

Home Create

username

Search

## Modifier un utilisateur

Nom d'utilisateur:

user1

Photo :

avatar.png

Sexe:

female

Email:

med.ham9@gmail.com

Dob:

01/05/2020

News:

☐

Modifier

### 2. Création d'un utilisateur :

localhost:3000/users

Home Create

username

Search

## Créer un utilisateur

Nom d'utilisateur:

Photo :

Sexe:

Email:

Dob:

News:

☐

### 3. Recherche par username :

localhost:3000/users/1/10?search=user

Home Create

user Search

## Liste des utilisateurs

Photo	Username	Sexe	Dob	News	Email	Action
	user1	female	05/01/2020	No	med.ham9@gmail.com	
	users2	male	04/01/2020	Yes	med@gmail.com	
	username	female	27/05/2019	No	med0@gmail.com	
	username65	male	02/02/2019	No	med0@gmail.com	
	user_over	female	05/01/2020	Yes	mai@idh.com	

1






4. Le tri par dob ou sexe :

Home Create

username

Search

## Liste des utilisateurs

Photo	Username	Sexe	Dob	News	Email	Action
	users2	male	04/01/2020	Yes	med@gmail.com	   
	medoman3	male	04/01/2020	No	med@gmail.com	   
	medoman4	male	04/01/2020	No	med@gmail.com	   
	medoman5	male	04/01/2020	No	med@gmail.com	   
	medoman6	male	15/02/2019	No	med@gmail.com	   
















1 2 3

Home Create

username

Search

## Liste des utilisateurs

Photo	Username	Sexe	Dob	News	Email	Action
	user1	female	05/01/2020	No	med.ham9@gmail.com	   
	username	female	27/05/2019	No	med0@gmail.com	   
	user_over	female	05/01/2020	Yes	mai@idh.com	   
	test1	female	05/01/2015	No	test@gmail.com	   
	users2	male	04/01/2020	Yes	med@gmail.com	   

1 2 3

Home Create

username

Search

## Liste des utilisateurs

Photo	Username	Sexe	Dob	News	Email	Action
	test1	female	05/01/2015	No	test@gmail.com	   
	username65	male	02/02/2019	No	med0@gmail.com	   
	medoman6	male	15/02/2019	No	med@gmail.com	   
	medoman9	male	16/02/2019	No	med@gmail.com	   
	medoman10	male	17/02/2019	No	med@gmail.com	   

1 2 3

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>