

## TP N°2 NoSQL : Modélisation orienté document

### Partie 1 : Installation de Mongo DB et Mongosh

1. Exécutez le fichier d'installation « mongodb-windows-x86\_64-7.0.3-signed.exe »
2. Suivez les instructions de l'assistant d'installation. Par défaut, MongoDB sera installé dans le répertoire **C:\Program Files\MongoDB**.
3. Cochez l'option "Install MongoDB Compass", pour installer l'interface graphique MongoDB Compass.
4. Cliquez sur "Next" et terminez l'installation.
5. Ajoutez path de mongoDB « C:\Program Files\MongoDB\Server\7.0\bin » Dans la barre de recherche, cherchez "**Système**" ensuite cliquez "**Paramètres système avancés**" → Choisissez "**Variables d'environnement**" ensuite dans "**Variables utilisateur**" cliquez sur "**Nouvelle**".
6. Insérez le dossier « mongosh-2.0.2-win32-x64 » sous le répertoire « C : »
7. Ajoutez path de mongosh « C:\mongosh-2.0.2-win32-x64\bin » dans la fenêtre de « modifier les variables d'environnements systèmes » → variables d'environnements
8. Testez l'installation de mongoDB et mongosh avec les commandes suivantes :
  - mongod --version
  - mongosh

Sous la fenêtre « invite de commande »

### Partie 2 : Travaux pratiques

#### Exercice 1

On souhaite gérer un club de tennis en ligne avec la possibilité de réserver un terrain à un créneau horaire. Le site ne gère que des réservations pour des matchs en simple. Voici la structure de la base de données :

#### Table joueurs

<u>id_joueur</u>	nom_joueur	prenom_joueur	login	mdp
1	Dupont	Alice	alice	1234
2	Durand	Belina	belina	5694
3	Caron	Camilia	camilia	9478
4	Dupont	Dorine	dorine	1347

#### Table matchs

<u>id_match</u>	date	id_creneau	id_terrain	id_joueur1	id_joueur2
1	2020-08-01	2	1	1	4
2	2020-08-01	3	1	2	3
3	2020-08-02	6	2	1	3

4	2020-08-02	7	2	2	4
5	2020-08-08	3	3	1	2
6	2020-08-08	5	2	3	4

### Table terrains

id_terrain	nom_terrain	surface
1	stade	terre battue
2	gymnase	synthétique
3	hangar	terre battue

### Table creneaux

id_creneau	plage_horaire
1	8h-9h
2	9h-10h
3	10h-11h
4	11h-12h
5	12h-13h
6	13h-14h
7	14h-15h
8	15h-16h
9	16h-17h
10	17h-18h
11	18h-19h
12	19h-20h

1. Créer la base de données « club\_tennis ».
2. Créez-la ou les collections et insérez les documents.
3. Ecrire une requête qui renvoie les prénoms des joueurs dont le nom est 'Dupont'.
4. Ecrire une requête qui modifie le mot de passe de Dorine Dupont, son nouveau mot de passe étant 1976.
5. Ecrire une requête permettant d'ajouter le nouveau membre « Zora MAGID » dont le login est « zora » et le mot de passe 2021.
6. Déterminer le jour et la plage horaire du match entre Durand Belina et Caron Camilia.
7. Déterminer le nom des deux joueurs qui sont les seuls à avoir joué dans le "hangar".
8. Compter le nombre de matchs joués sur chaque terrain.
9. Trouver la plage horaire la plus fréquemment utilisée pour les matchs.
10. Trouver le joueur qui a joué le moins de matchs.
11. Trouver le terrain le plus utilisé pour les matchs.

## Exercice 2 : Modèles NoSQL (Modélisation orientée documents)

On donne la représentation textuelle simplifiée d'une base de données concernant un cycle de formation destiné à des étudiants. Il regroupe un ensemble de matières. On considère que chaque enseignant n'enseigne qu'une seule matière et qu'à la fin du cycle de formation, une note par matière, est attribuée à chaque étudiant. D'autre part, les étudiants peuvent ne pas suivre les mêmes matières.

- ETUDIANT(NEtudiant, Nom, Prénom)
- MATIERE(CodeMat, LibelléMat, CoeffMat)
- ENSEIGNANT(CodeEns, NomEns, GradeEns, #CodeMat)
- Note(#NEtudiant, #CodeMat, Date, Note)

### ETUDIANT

NumEtudiant	Nom	Prenom	Section
01234567	bruno	Alex	Informatique
01234568	lavand	Stive	Math
01234569	lahoche	Aline	Informatique

### MATIERE

CodeMat	NomMat	Coefficient
12508	Base de données	1.5
12518	Algorithmes	2

### NOTE

NumEtudiant	CodeMat	Note_examen
01234567	12508	15.5
01234567	12518	5.5
01234568	12518	10.5
01234569	12518	8.75

### ENSEIGNANT

CodeEns	NomEns	GradeEns	CodeMat
123	Lois	Grd1	12508
124	Philippe	Grd2	12518

Exprimez-en NoSQL les requêtes suivantes :

1. Créer la base de données « cycle\_formation ».
2. Créez-la ou les collections et insérez les documents.
3. Quel est le nombre total d'étudiants ?
4. Quelles sont, parmi l'ensemble des notes, la note la plus haute et la note la plus basse ?
5. Quelles sont les moyennes de chaque étudiant dans chacune des matières ?
6. Quelle est la moyenne générale de chaque étudiant ?
7. Les numéros des étudiants dont la moyenne entre 7 et 12.
8. La liste des étudiants dont le nom commence par 'l'.
9. Le nombre des étudiants qui ont comme matière '12518'.
10. La somme des coefficients des matières.
11. Les noms des étudiants qui une note d'examen >10.
12. Afficher les noms et les coefficients des matières étudiées par l'étudiant "01234568".

13. Afficher les informations relatives aux étudiants (Code, Nom et Date de naissance) selon l'ordre alphabétique croissant du nom
14. Afficher les noms et les grades des enseignants de la matière dont le nom est 'BD'.
15. Afficher la liste distincte formée des noms et les coefficients des différentes matières qui sont enseignées par des enseignants de grade 'Grd2'.
16. Afficher le nombre d'enseignants de la matière dont le nom est 'Informatique'

## Travail à rendre : compte rendu

Le compte rendu contient :

- Question demandée.
  - Code de requête de question demandée.
  - Imprime écran de résultat de requête.
- 

## Projet

### 1. Description

- ✓ On se propose de développer une application qui permet de présenter les départements d'une société informatique.

**app** : le composant parent incluant les composants suivants :

- **accueil** : afficher le message de bienvenue et une petite description de la société.
- **apropos**: afficher les informations contact de la société
  - nom : nom de la société
  - email : email de la société
  - Téléphone
  - faxe
  - logo : image
  - .....
- **listDeparts** :
  - formAjout : formulaire d'ajout du nouveau département
  - Affichage des différents départements avec possibilité de modification et suppression.



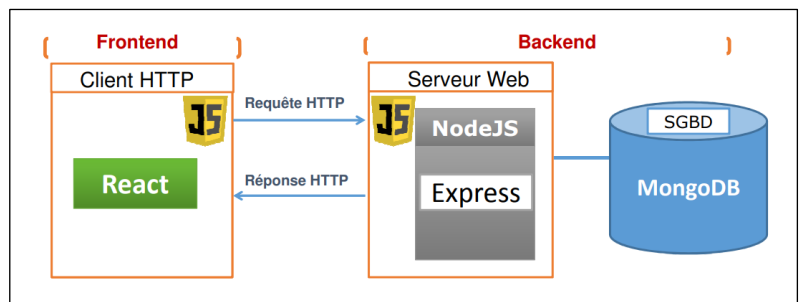
✓ Une base de données locale, ayant la collection 'départements'.

- Tout en gardant les identifiants générés par défaut `_id`, insérer au niveau de cette collection, les documents suivants :

```
[{ code : 1, nom : "Commercial" },  
{ code : 2, nom : "Développement"},  
{ code : 3, nom : "Maintenance"}  
.....]
```

## 2. Outil

- Visual studio code
- Node JS
- React
- Express JS
- MongoDB et Mongo Atlas
- mongoose



## 3. Travaux demandés :

- Créer les « composants ».js (accueil, apropos ,...)
- Assurer la navigation entre les composants.
- Mise en place du serveur avec express
- Créer et connecter à la base de données
- Connecter à la base de données et APIs avec Express JS
- Récupérer et afficher, ajouter, mettre à jour et supprimer la liste des départements enregistrées au niveau de la base de données
- Assurer l'interaction du serveur avec l'application FrontEnd React JS

## 4. Installation de visual studio code + node js

1. Exécutez le fichier d'installation « VSCodeUserSetup-x64-1.84.2.exe ».
2. Exécutez le fichier d'installation « node-v20.10.0-x64.exe ».
3. Pour vérifier les versions installées de node js :

- `node --version` ou `node -v`
- `npm --version` ou `npm -v`

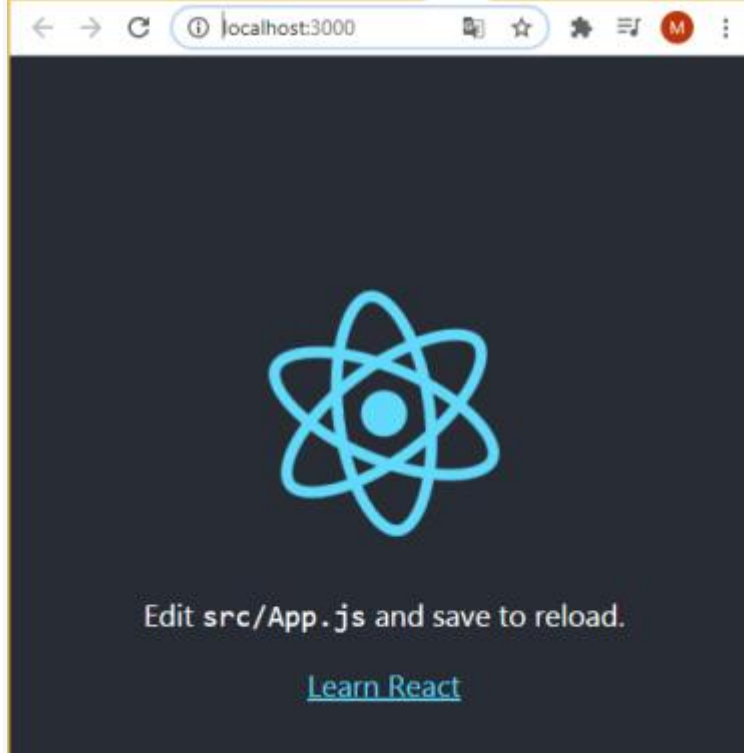
4. Créez un dossier « ma-societe » sous le répertoire « D: » .
5. Installez l'outil : create-react-app :

- `npm install -g create-react-app`

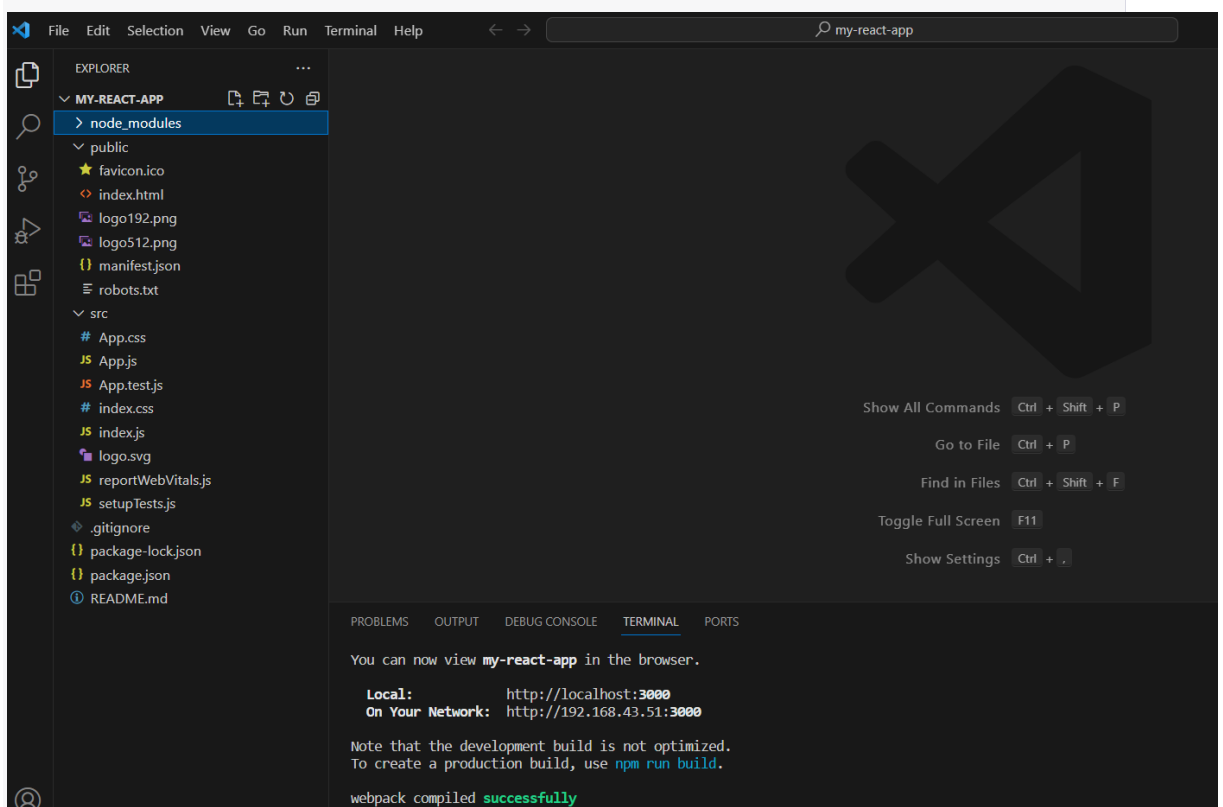
6. Créez un nouveau dossier « D:/ ma-societe »

- `npx create-react-app my-react-app`
- `cd my-react-app`
- `npm start`

7. L'application se lance par défaut en local sur : <http://localhost:3000/>



8. Accédez au dossier «my-react-app» via l'outil Visual studio code.



9. Commencez votre projet.