

RAPPORT

La mise en œuvre de ce projet a été divisé en 2 parties qui sont :

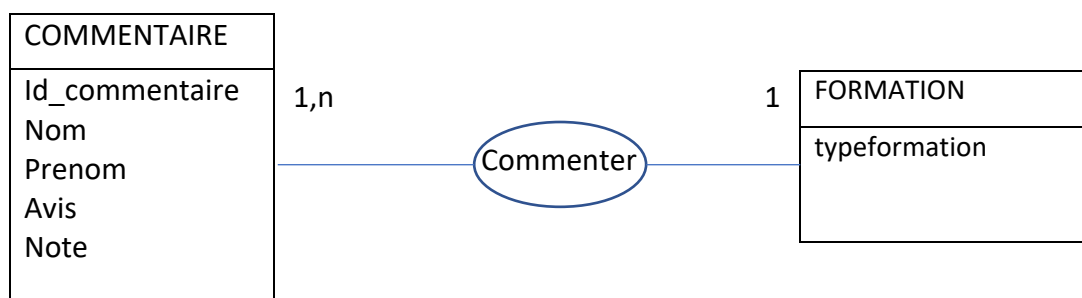
- La conception de la base de données
- Le développement

Conception de la base de données

Pour la conception de la base, j'ai sollicité l'utilisation du logiciel « workbench » pour la conception du schéma logique au schéma physique.

MCD

Voici le modèle conceptuel de données



Le MCD sert à voir une vue d'ensemble et globale de la base de donnée , c'est le point de départ pour avoir le modèle logique et puis le modèle physique .

Dictionnaire de données

Attribut	Types
Id_commentaire	entier
Nom	Caractère
Prenom	Caractère
Avis	text
Note	entier
typeformation	caractère

Il est très utile de savoir que l'attribut typeinformation est à la fois la clé primaire du table formation. Le fait qu'il soit de type caractère joue un rôle très important dans le code

Diagramme

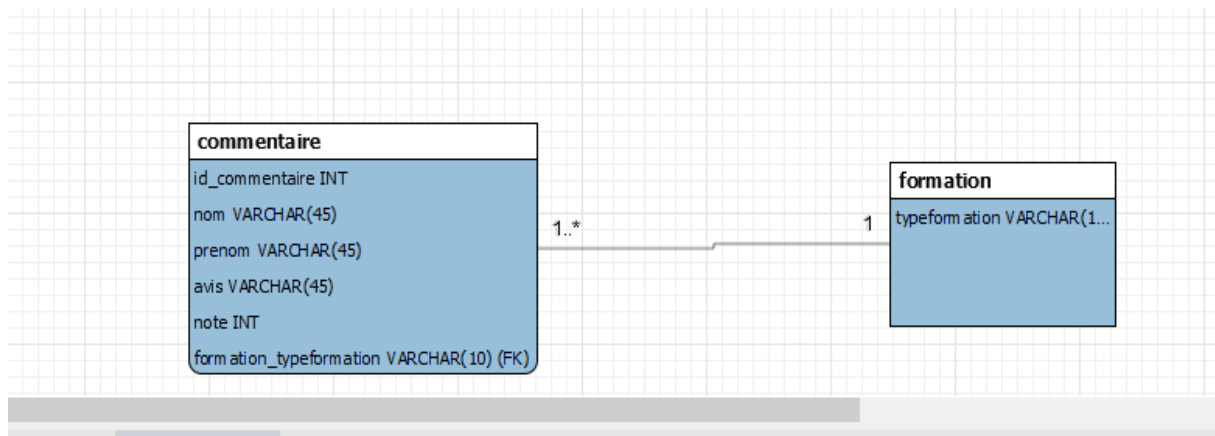


Figure 1: Diagramme de classe

Voilà donc le modèle logique de données de notre base de données, avec deux tables « commentaire et formation » pour une cardinalité 1-1n.

En passant le mode forwarding du database j'ai pu générer le schéma physique.

Le typeformation est composé de :

- Frontend
- Backend
- Ux
- Marketing

Développement

Outils

- Visual Studio Code
- Wamp server

Package list

- NPM
- Express
- Body-parser
- Ejs

Ce projet a été développé en suivant la norme du MVC qui est Model-Views-Controller

Dans le dossier Model on trouve le fichier db.js qui contient les scripts pour la connexion à la base de données.

Le views sert pour l'affichage ou rendu des pages.

La création du dossier controller n'a pas été pertinente pour ce projet

Dans la page index.js se trouve le cœur du projet car c'est dedans qu'on trouve les routes des API et les requêtes pour l'insertion et l'affichage de la base.

- **Connection à la base de données**

```
models > JS db.js > ...
1  const mysql = require('mysql');
2
3
4  //console.log(process.env)
5
6
7  const connection= mysql.createConnection({
8      host: '127.0.0.1',
9      user: 'root',
10     password: '',
11     database: 'base',
12     port: '3309'
13 })
14
15
16
17
18 connection.connect((err)=>{
19     if(!err)
20         console.log("connection etabli")
21     else
22         console.log('connection failed' + JSON.stringify(err,undefined,2))
23 })
24
25 module.exports=connection;
26
```

Figure 2: connexion à la base de données

NB : pour une raison inconnue l'environnement « .env » ne marche pas du coup , je suis obligé de ne pas utiliser cette variable d'environnement .

Le Module.exports = connection est utilisé pour que le script dans la variable connection soit vu dans d'autre page.

- **Création de route**

```
app.get('/marketing', function(req,res){
  res.render('marketing')
})
```

```
app.get('/backend', function(req,res){
  res.render('backend')
})
```

```
app.get('/contact', function(req,res){
  res.render('contact')
})
```

Figure 3: route

- Requête d'insertion

```
const insertquery = "INSERT INTO commentaire (nom, prenom, avis, note, formation_typeformation) VALUES (?, ?, ?, ?, ?)"
```

- Requête pour l'envoi de donnée

```
class comment {
  static create(firstname, lastname, avis, note, formation_typeformation, query, callback) {
    connection.query(query, [firstname, lastname, avis, note, formation_typeformation], function(err, result) {
      if (err) throw err
      callback(result)
    })
  }
}

if(req.body.nom=="" || req.body.prenom=="" || req.body.avis=="" || req.body.note=="") {
  res.send('vous devez remplir tout les formulaire')
  res.redirect('/contact')
}
else{
```

```
comment.create(req.body.nom, req.body.prenom, req.body.avis, req.body.note
, req.body.formation, insertquery, function(){
    console.log(req.body.nom, req.body.prenom, req.body.avis, req.body.note
, req.body.formation )

    res.redirect('/contact')
});
```

Le package EJS est très utilisé pour faire le Views dans nodeJS. Le views sera le remplacement de HTML .
et le bodyParser récupère les balise HTML qui se trouve dans le fichier .EJS

```
app.set('view engine', 'ejs');
```

Résumé

Pour le front-end des modifications ont été apporter pour régler quelque imperfection, tant au niveau du balise HTML que CSS.

Le triage du meilleur note par formation a été le but principal de ce projet