

RAPPORT DE STAGE

Développeur web

HOMMET Milan

Du 3 janvier au 11 février

Tuteur de stage : Sophie Schmitzler – Professeure de vente

Enseignant référant : Sébastien Gaulier – Professeur de BTS SIO

Établissement / Formation : Établissement Saint Adjutor –
BTS Services informatiques aux organisations (SIO) - Option
SLAM

Établissement d'accueil : Ensemble scolaire catholique –
Saint Agnès / Jeanne d'Arc

Sommaire

PRESENTATION DU PROJET PROFESSIONNEL	3
PRESENTATION DE L'ESC ST AGNES – JEANNE D'ARC.....	4
PRESENTATION DU PROJET	5
1. Moyens mis en place.....	5
2. Organisation de travail.....	5
3. Interaction avec les utilisateurs	5
TÂCHES REALISEES	6
1. Création de la base de données.....	6
2. Démarrage du projet.....	6
3. Menu de navigation et routes.....	7
4. Page d'accueil	8
5. Analyse des fonctionnalités	10
6. Page de connexion et permissions.....	19
7. Page de contact.....	20
8. Documentation	21
CONCLUSION.....	22
1. Ce que j'ai appris.....	22
2. Remerciements	22

PRESENTATION DU PROJET PROFESSIONNEL

Disposant d'un baccalauréat ES option sciences politiques mention européenne, je suis actuellement en deuxième année de **BTS SIO** (Services Informatiques aux Organisations) et j'ai choisi l'option **SLAM** (Solutions Logicielles et Applications Métiers).

Je souhaite à la suite de cette formation, faire une troisième année en bachelor dans une école de développement web afin de me spécialiser dans ce domaine.

Pour ce stage j'ai utilisé principalement le Framework React et le langage Javascript.

Mon but est de travailler dans le développement web et de devenir développeur fullstack (HTML, CSS, JAVASCRIPT, NodeJS, MySQL...).

PRESENTATION DE L'ESC ST AGNES – JEANNE D'ARC



Ensemble Scolaire Catholique
— Sainte Agnès - Jeanne d'Arc —

Le collège Jeanne d'Arc et le lycée Saint Agnès sont des établissements scolaires privés dans le centre de Vernon.

Le lycée Saint Agnès est un lycée privé situé au 126 rue d'Albufera qui propose quatre Bac pro : commerce, vente, accueil-relation clients et usagers et gestion-administration ainsi que deux CAP : employé commerce multispécialité et métiers de la blanchisserie. Il propose aussi deux BTS : MCO et NDRC.



PRESENTATION DU PROJET

Ce stage s'est déroulé dans le cadre d'un projet chef d'œuvre consistant à retrouver et contacter les anciens élèves du lycée afin de récupérer des informations comme le poste occupé et l'entreprise actuelle, l'adresse, le département...

Mon travail au sein de ce projet était donc de développer le site permettant d'enregistrer les informations récoltées et de les modifier, supprimer ou consulter par filtres d'années, de nom, de département ou d'entreprise.

1. Moyens mis en place

Pour réaliser ce site, j'ai utilisé le Framework React et j'ai utilisé du HTML, CSS et principalement du Javascript.

2. Organisation de travail

J'ai travaillé principalement en télétravail mais aussi en présentiel les mardis et vendredis après-midi. Pour planifier mes tâches restantes j'ai utilisé Microsoft To Do afin de savoir ce qu'il me restait à faire au fur et à mesure de mon avancement.

3. Interaction avec les utilisateurs

Tous les mardis après-midi, je suis allé dans une classe de BAC Pro Vente pour montrer mes avancements aux élèves et professeurs et recueillir leurs idées et les points d'améliorations. Ces interactions avec les utilisateurs ont été enrichissantes car elles permettaient d'avoir des points de vue extérieurs sur l'ergonomie ou le design du site par exemple.

TÂCHES REALISEES

1. Création de la base de données

J'ai créé un MCD de la base de données et j'ai réfléchi aux différentes tables à créer puis j'ai créé la base de données dans phpMyAdmin avec MySQL.

J'ai créé les éléments des tables en fonction de la fiche de contact qui avait été imaginée par les élèves.

J'ai mis en relation les tables élèves, entreprises et secteurs avec des clés primaires et clés étrangères.

eleId	elePrenom	eleNom	eleGenre	eleNumTel	eleMail	eleAdresse	eleDep	eleEntrepriseId	elePosteOccupe	eleClasse	eleAnneeLycee
1	testPrenom	testNom	h	603963928	awd@gmail.com	tes	27	3	Test	TES3	2010
9	Antoine	Auvray	H	1234567890	test45@gmail.com	test adresse maj	27	1	Gestionnaire	TES2	2019
12	milan	test5	F	1234567890	milan.hommet@protonmail.com	22d sente de slares 27200 vernon	76	1	Vendeur	TES3	2006
13	Milan	Hommet	H	603963928	milan.hommet@protonmail.com	22d sente des lares	76	1	Vendeur	TES3	2018
14	Milan	Hommet	H	603963928	milan.hommet@protonmail.com	22d sente des lares	14	1	Vendeur	TES3	2018

ected: Edit Copy Delete Export

25 Filter rows: Search this table Sort by key: None

rd Export Display chart Create view

2. Démarrage du projet

J'ai créé un dossier pour ce stage et j'ai fait npm init dedans afin de créer un nouveau projet React. J'ai ensuite fait le tri dans les fichiers et enlevé ceux dont je n'avais pas l'utilité et j'ai commencé à créer le site.

3. Menu de navigation et routes

J'ai créé le menu afin de naviguer entre les pages du site :

```
1 .menu {
2   background: rgb(63, 1, 90);
3   height: 4rem;
4
5 }
6 .menu ol {
7   list-style-type: none;
8   margin: 0 auto;
9   padding: 0;
10 }
11 .menu > ol {
12   max-width: 1000px;
13   padding: 0 2rem;
14   display: flex;
15 }
16 .menu > ol > .menu-item {
17   flex: 1;
18   padding: 0.75rem 0;
19 }
20 .menu > ol > .menu-item:after {
21   content: "";
22   position: absolute;
23   width: 4px;
24   height: 4px;
25   border-radius: 50%;
26   bottom: 5px;
27   left: calc(50% - 2px);
28   background: #e41c54;
29   will-change: transform;
30   transform: scale(0);
31   transition: transform 0.2s ease;
32 }
```

Code CSS

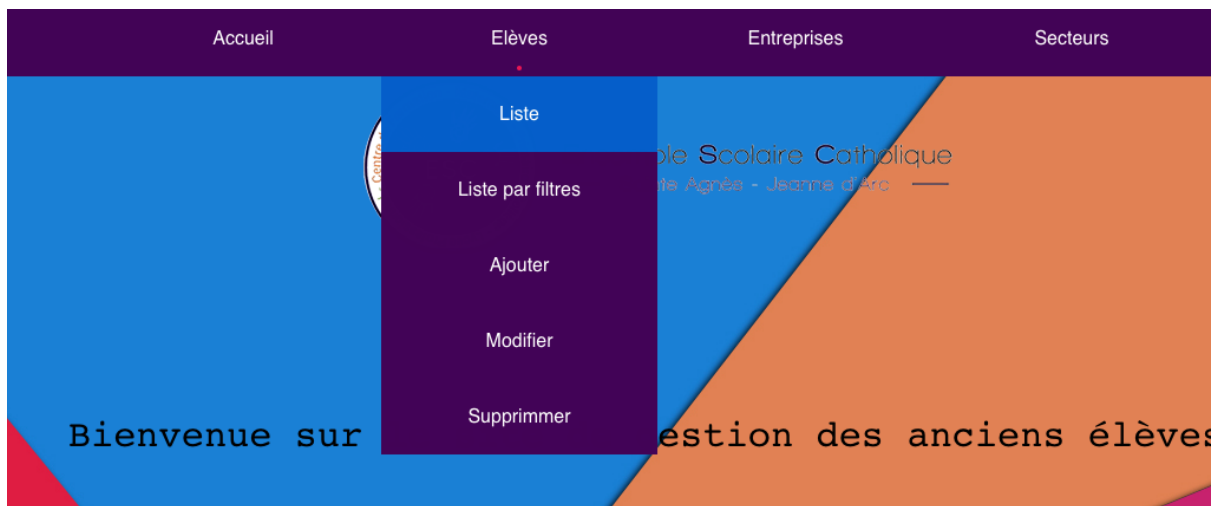
```
1 {elevelist.map((val, key) => {
2   return (
3
4 <div class="at-column">
5   <div class="at-user">
6     <div class="at-user_name"><h1>{val.elePrenom} {val.eleNom}</h1></div>
7     <div class="at-user_title"><label>Genre :</label> {val.eleGenre}</div>
8     <div class="at-user_title"><label>Tél :</label> {val.eleNumTel}</div>
9     <div class="at-user_title"><label>Mail :</label> {val.eleMail}</div>
10    <div class="at-user_title"><label>Adresse :</label> {val.eleAdresse}</div>
11    <div class="at-user_title"><label>Département :</label> {val.eleDep}</div>
12    <div class="at-user_title"><label>Id Entreprise :</label> {val.eleEntrepriseId}</div>
13    <div class="at-user_title"><label>Poste :</label> {val.elePosteOccupe}</div>
14    <div class="at-user_title"><label>Classe :</label> {val.eleClasse}</div>
15    <div class="at-user_title"><label>Année du diplôme :</label> {val.eleAnneelycee}</div>
16  </div>
17 </div>
```

Code html

J'ai ensuite créé les routes permettant de relier les boutons du menu aux pages en utilisant routes et BrowserRoutes :

```
1 <BrowserRouter>
2   <Routes>
3     <Route path="/" element={<Navbar />} />
4     <Route index element={<LoginPage />} />
5     <Route path="home" element={<Home />} />
6     <Route path="login" element={<LoginPage />} />
7     <Route path="list" element={<List />} />
8     <Route path="listFilter" element={<ListFilter />} />
9     <Route path="add" element={<Add />} />
10    <Route path="update" element={<Update />} />
11    <Route path="delete" element={<Delete />} />
12    <Route path="entreList" element={<EntreList />} />
13    <Route path="entreAdd" element={<EntreAdd />} />
14    <Route path="entreDelete" element={<EntreDelete />} />
15    <Route path="secteurList" element={<SecteurList />} />
16    <Route path="secteurAdd" element={<SecteurAdd />} />
17    <Route path="secteurDelete" element={<SecteurDelete />} />
18  </Routes>
19 </BrowserRouter>
```

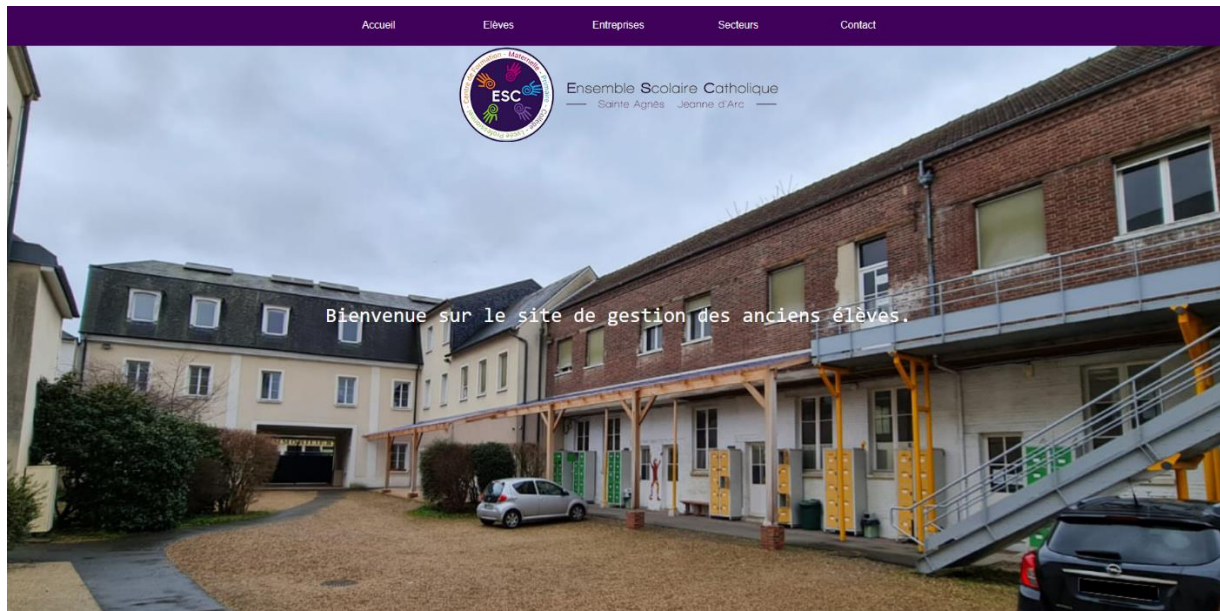
Routes dans App.js



Résultat final

4. Page d'accueil

J'ai créé la page d'accueil du site en mettant un fond, le logo de l'établissement et un texte animé qui accueille le visiteur :



Résultat final


```

1  import React from "react";
2  import "./Home.css";
3
4  export default function Home() {
5    return (
6      <>
7        <div className="home">
8          <div id="logo"></div>
9
10         <div class="wrapper">
11           <div class="typing-demo">
12             Bienvenue sur le site de gestion des anciens élèves.
13           </div>
14         </div>
15       </div>
16     </>
17   )
18 }

```

Component Home

```

1  .wrapper {
2    height: 50vh;
3    display: flex;
4    align-items: center;
5    justify-content: center;
6  }
7  .typing-demo {
8    width: 52ch;
9    animation: typing 2s steps(22), blink .5s step-end infinite alternate;
10   white-space: nowrap;
11   overflow: hidden;
12   border-right: 3px solid;
13   font-family: monospace;
14   font-size: 2em;
15   color: black;
16 }

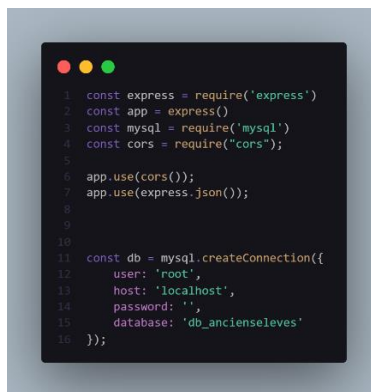
```

Code CSS de l'accueil

5. Analyse des fonctionnalités

- Consultation des élèves, des entreprises et des secteurs

Pour accéder à la base de données et procéder à la consultation, l'ajout, la modification et la suppression, j'ai créé une API permettant de se connecter à la base de données MySQL et d'exécuter des requêtes.



```
1 const express = require('express')
2 const app = express()
3 const mysql = require('mysql')
4 const cors = require('cors');
5
6 app.use(cors());
7 app.use(express.json());
8
9
10
11 const db = mysql.createConnection({
12   user: 'root',
13   host: 'localhost',
14   password: '',
15   database: 'db_ancienneleves'
16 });
```

*Importation des dépendances et connexion
à la base de données*

J'ai créé les requêtes permettant d'obtenir la liste des élèves, des entreprises et des secteurs :



```
1 app.get('/eleves', (req, res) => {
2   db.query("SELECT * FROM eleves", (err, result) => {
3     if (err) {
4       console.log(err);
5     } else {
6       res.send(result);
7     }
8   });
9 });
```

Ex : Requête de sélection des élèves

L'API retourne sur le localhost:3001/eleves ou /entreprises ou /secteurs la liste des données récupérées :

```
[{"eleId":14,"elePrenom":"Céline","eleNom":"Duthil","eleGenre":"F",
"pots","eleDep":17,"eleEntrepriseId":4,"elePosteOccupe":"Nourrice"},
{"eleId":20,"elePrenom":"Frederic","eleNom":"Boissard","eleGenre":"F",
"Quevilly","eleDep":76,"eleEntrepriseId":5,"elePosteOccupe":"Agent",
{"eleId":30,"elePrenom":"Charlène","eleNom":"Payen","eleGenre":"F",
30340","eleDep":30,"eleEntrepriseId":6,"elePosteOccupe":"Auxiliaire",
{"eleId":31,"elePrenom":"Guillaume","eleNom":"Dhorothée","eleGenre":"F",
Evreux","eleDep":27,"eleEntrepriseId":7,"elePosteOccupe":"Responsable",
{"eleId":32,"elePrenom":"Kathy","eleNom":"Alexandre","eleGenre":"F",
78270","eleDep":78,"eleEntrepriseId":8,"elePosteOccupe":"Chef de projet",
{"eleId":33,"elePrenom":"Anthony","eleNom":"Koupp","eleGenre":"H",
27120","eleDep":27,"eleEntrepriseId":9,"elePosteOccupe":"Commercial"}
```

Il a ensuite fallu mettre en forme et afficher les données sur le site. J'ai donc utilisé Axios pour récupérer les données et les mettre dans un tableau :

```
1
2 var [elevelist, setElevelist] = useState([]);
3
4 const getEleves = () => {
5   Axios.get("http://localhost:3001/eleves").then((response) => {
6     setElevelist(response.data);
7   });
8 };
9
10
11 useEffect(() => {
12   getEleves()
13 });
```

Enfin, j'ai affiché ces données dans des cartes auxquelles j'ai ajouté du style et une animation de survol :

```
1 {elevelist.map((val, key) => {
2   return (
3
4   <div class="at-column">
5     <div class="at-user">
6       <div class="at-user__name"><h1>{val.elePrenom} {val.eleNom}</h1></div>
7       <div class="at-user__title"><label>Genre :</label> {val.eleGenre}</div>
8       <div class="at-user__title"><label>Tél :</label> {val.eleNumTel}</div>
9       <div class="at-user__title"><label>Mail :</label> {val.eleMail}</div>
10      <div class="at-user__title"><label>Adresse :</label> {val.eleAdresse}</div>
11      <div class="at-user__title"><label>Département :</label> {val.eleDep}</div>
12      <div class="at-user__title"><label>Id Entreprise :</label> {val.eleEntrepriseId}</div>
13      <div class="at-user__title"><label>Poste :</label> {val.elePosteOccupe}</div>
14      <div class="at-user__title"><label>Classe :</label> {val.eleClasse}</div>
15      <div class="at-user__title"><label>Année du diplôme :</label> {val.eleAnneelYcee}</div>
16    </div>
17  </div>
18 )}
```

Le « *elevelist.map((val, key) »* permet d'afficher une carte pour chaque élève.



Partie du code CSS de la liste des élèves

Céline Duthil	Frederic Boissard	Charlène Payen
Genre : F	Genre : H	Genre : F
Tél : 617803677	Tél : 661930577	Tél : 666936709
Mail : celleduthil1983@gmail.com	Mail : fboissard@sna-vernon27.fr	Mail : charlene.P74@hotmail.fr
Adresse : 14 chemin du bois de la croix 17000 La Chapelle des pots	Adresse : 20 rue de prevoyants appartement N°9 batiment 3 mat 76120 Le Grand Quevilly	Adresse : 14 route canabias 30340
Département : 17	Département : 76	Département : 30
Id Entreprise : 4	Id Entreprise : 5	Id Entreprise : 6
Poste : Nourrice	Poste : Agent d'animation	Poste : Auxiliaire petite enfance
Classe : CAP VENTE	Classe : BAC PRO VR	Classe : BAC PRO VENTE
Année du diplôme : 1999	Année du diplôme : 2002	Année du diplôme : 2007

Résultat final de la liste des élèves

J'ai reproduit le même cheminement pour la liste des entreprises et des secteurs en adaptant les requêtes :

Liste des entreprises					
4 Nourrice Adresse: 8 domicile Id du secteur: 3	5 Mairie de Vernon Adresse: Pl Adolphe Barette 27200 Vernon Id du secteur: 4	6 ALM Crèche Adresse: 12 impasse du grand chêne 30100 Alès Id du secteur: 3	3 Garde d'enfants	4 Animation	5 Batiment
7 MPO Fenêtre Adresse: 76000 Rouen Id du secteur: 5	8 Orange Palaiseau Adresse: 24 rue Emile Sauvalat 91120 Palaiseau Id du secteur: 6	9 France boisson Adresse: 29 rue de buchelay 76110 Rosny-sur-seine Id du secteur: 7	6 Télécommunications	7 Commerce	

- Ajout d'élèves, d'entreprises et de secteurs

J'ai mis en place la possibilité d'ajouter un élève, une entreprise ou un secteur :

Tout d'abord, j'ai créé les requêtes nécessaires dans mon API :

```
1 app.post('/create', (req, res) => {
2   const prenom = req.body.prenom;
3   const nom = req.body.nom;
4   const genre = req.body.genre;
5   const numTel = req.body.num;
6   const eMail = req.body.eMail;
7   const adresse = req.body.adresse;
8   const departement = req.body.departement;
9   const entrepriseId = req.body.entrepriseId;
10  const posteOccupe = req.body.posteOccupe;
11  const classe = req.body.classe;
12  const anneeLycee = req.body.anneeLycee;
13
14  db.query('INSERT INTO eleves (elePrenom, eleNom, eleGenre,
15    [prenom, nom, genre, numTel, eMail, adresse, departement,
16    (err, result) => {
17      if (err) {
18        console.log(err)
19      } else {
20        res.send('Values Inserted')
21      }
22    }
23  });
24 });
```

Insertion d'un élève

```
1 const [prenom, setPrenom] = useState("");
2 const [nom, setNom] = useState("");
3 const [genre, setGenre] = useState("");
4 const [num, setNum] = useState("");
5 const [eMail, setEmail] = useState("");
6 const [adresse, setAdresse] = useState("");
7 const [departement, setDepartement] = useState("");
8 const [entrepriseId, setEntrepriseId] = useState("");
9 const [posteOccupe, setPosteOccupe] = useState("");
10 const [classe, setClasse] = useState("");
11 const [anneeLycee, setAnneeLycee] = useState("");
```

*Déclaration de tableaux permettant de stocker
les champs remplis.*

```

1  const addEleve = () => {
2    Axios.post("http://localhost:3001/create", {
3      prenom: prenom,
4      nom: nom,
5      genre: genre,
6      num: num,
7      eMail: eMail,
8      adresse: adresse,
9      departement: departement,
10     entrepriseId: entrepriseId,
11     posteOccupe: posteOccupe,
12     classe: classe,
13     anneeLycee: anneeLycee,
14   }).then(() => {
15     alert("Élève ajouté avec succès");
16     setEleveList([
17       ...eleveList,
18       {
19         prenom: prenom,
20         nom: nom,
21         genre: genre,
22         num: num,
23         eMail: eMail,
24         adresse: adresse,
25         departement: departement,
26         entrepriseId: entrepriseId,
27         posteOccupe: posteOccupe,
28         classe: classe,
29         anneeLycee: anneeLycee,
30       },
31     ]);
32   });
33 };
34

```

Fonction addEleve permettant de récupérer les champs remplis dans les tableaux correspondants et de les envoyer en paramètres de la requête d'insertion de l'API.

```

1  <label for="eleves">Prenom:</label>
2    <input type="text" onChange={(event) => {
3      setPrenom(event.target.value);
4    }}
5  />

```

Champ à remplir qui ajoute le prénom entré dans le tableau du nom de l'élève.

Pour l'entreprise, j'ai créé un composant permettant de récupérer la liste des entreprises et de remplir une liste déroulante avec les noms des entreprises récupérées.

```

1  async getOptions() {
2    const response = await Axios.get('http://localhost:3001/entreprises')
3    const data = response.data
4
5    const options = data.map(d => ({
6      "value": d.entreId,
7      "label": d.entreNom
8    }))
9
10   this.setState({selectedOptions:options})
11 }

```

Récupération des entreprises et remplissage du tableau.

```

1  render() {
2    const entreValue = this.state.id;
3    return(
4      <div className='entreprises'>
5        <Select placeholder="Liste des entreprises..." options={this.state.selectedOptions} onChange={this.handleChange.bind(this)} />
6        <p>Numéro de l'entreprise : {this.state.id}</p>
7      </div>
8    )
9  }
10 }

```

Alimentation de la liste déroulante

Entreprise:

Ajouter une entreprise

Liste des entreprises... ▼

Nourrice

Mairie de Vernon

ALM Crèche

MPO Fenêtre

Orange Palaiseau

France boisson

Nom:

J'ai aussi ajouté une popup permettant d'ajouter une entreprise pendant l'ajout d'un élève :

Ajouter une entreprise:

Nom:

Adresse:

Secteur:

Liste des secteurs... ▼

Numéro du secteur :

Entrez le numéro du secteur...

Ajouter

- Modification des élèves

J'ai mis en place la possibilité de mettre à jour certaines données d'élèves en cas d'erreur ou de données obsolètes.

J'ai donc créé les requêtes d'update pour chaque champ à mettre à jour (ex : mail, numéro de téléphone...) :

```

1  app.put("/updateTel", (req, res) => {
2    const eleId = req.body.eleId;
3    const eleNumTel = req.body.eleNumTel;
4    db.query(
5      "UPDATE eleves SET eleNumTel = ? WHERE eleId = ?",
6      [eleNumTel, eleId],
7      (err, result) => {
8        if (err) {
9          console.log(err);
10       } else {
11         res.send(result);
12       }
13     }
14   );
15 });

```

```

1 const updateEleveNumTel = (eleId) => {
2   Axios.put("http://localhost:3001/updateTel", {eleNumTel: newNumTel, eleId: eleId }).then((response)=>{
3     alert("Numéro de téléphone mis à jour avec succès");
4   });
5 };

```

Fonction Axios permettant de mettre à jour le numéro de téléphone avec comme paramètre le nouveau numéro entré.

```

1 <div class="at-user__title"><label>Numéro de téléphone :</label> {val.eleNumTel}</div>
2   <div className="maj">
3     <input type="text" placeholder="Nouveau numéro de téléphone..." onChange={(event)=>
4     </div>

```

Code html / javascript de l'interface graphique

```

1 .container {
2   max-width: 1500px;
3   width: 100%;
4   margin: 50px auto;
5   box-sizing: border-box;
6   transform-origin: top center;
7   transform: scale(0.8);
8 }
9
10 .maj input {
11   width: 200px;
12   height: 25px;
13 }
14
15 .maj button {
16   width: 90px;
17   height: 40px;
18 }

```

J'ai créé une interface graphique permettant d'entrer la nouvelle valeur souhaitée dans un champ libre et de valider la mise à jour en cliquant sur un bouton.

Département : 17

Numéro de l'entreprise : 4

Poste : Nourrice

Classe : CAP VENTE

Année du diplôme : 1999

- Suppression des élèves, entreprises, secteurs

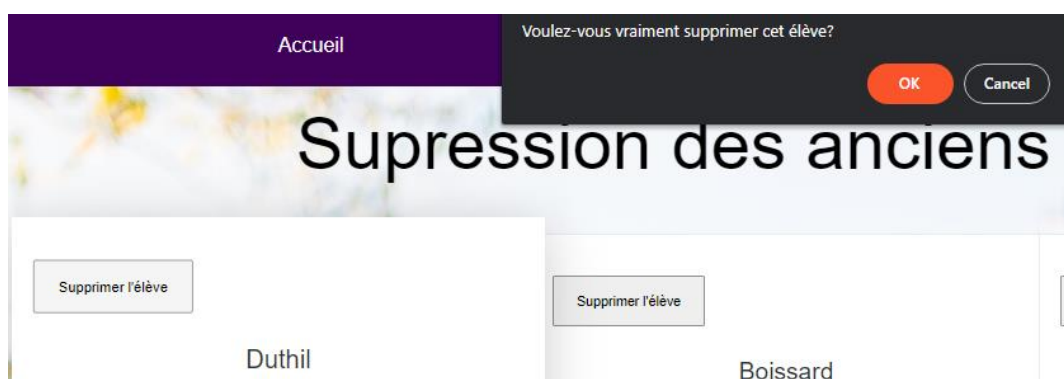
Pour la suppression j'ai d'abord réalisé les requêtes nécessaires dans mon API :

```
app.delete("/delete/:eleId", (req, res) => {
  const eleId = req.params.eleId
  db.query("DELETE FROM eleves WHERE eleId = ?", eleId, (err, result) => {
    if (err) {
      console.log(err);
    } else {
      res.send(result);
    }
  });
});
```

J'ai ensuite créé une page qui affiche la liste des élèves avec un bouton supprimer qui appelle une fonction qui envoie une requête de suppression dans mon API avec comme paramètre l'ID de l'élève à supprimer :

```
const deleteEleve = (eleId) => {
  Axios.delete(`http://localhost:3001/delete/${eleId}`).then((response) => {
    setEleveList(eleveList.filter((val) => {
      return val.eleId !== eleId
    })))
  })
};
```

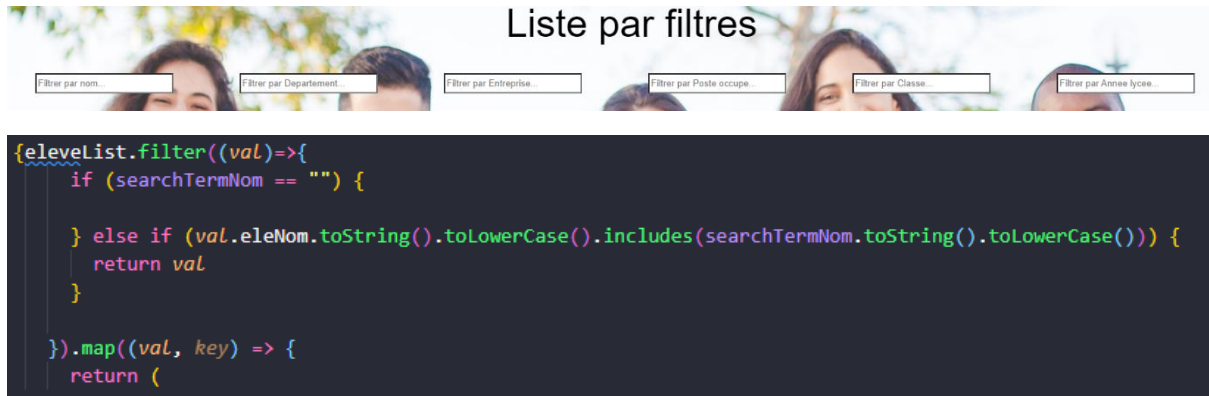
J'ai aussi ajouté une popup de validation pour éviter une suppression accidentelle.



Et j'ai reproduit la même chose en adaptant les requêtes et fonctions pour les entreprises et les secteurs.

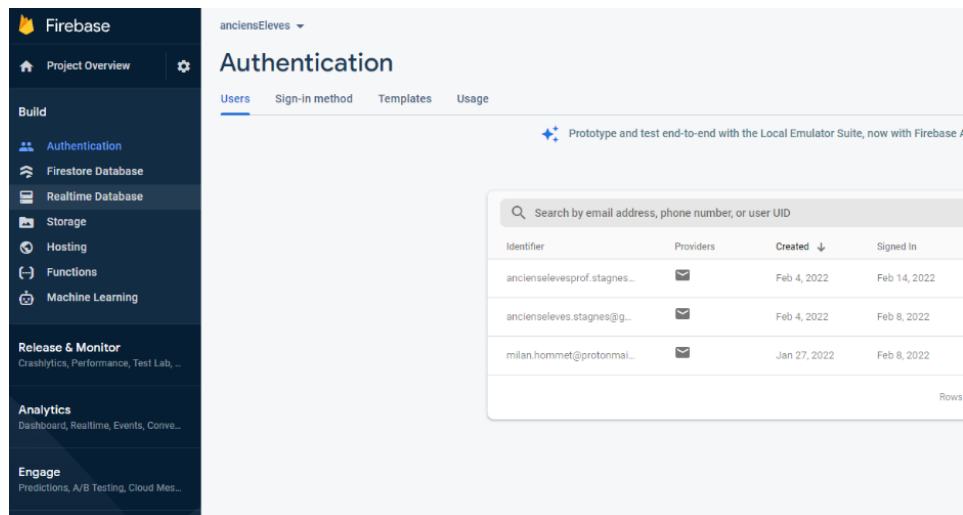
- Recherche des élèves par filtres

Enfin, j'ai mis en place une recherche par filtre (nom, département, entreprise, poste occupé, classe, année d'obtention du diplôme) en reprenant la page de la liste des élèves mais en ajoutant des champs de texte à remplir qui appliquent un filtre sur la liste :



6. Page de connexion et permissions

Pour l'authentification j'ai utilisé le service Firebase de Google ce qui m'a permis de créer un compte élève et un compte professeur.



J'ai créé un fichier Firebase-config afin de connecter mon projet à Firebase puis j'ai créé une page de connexion :



Au début de chaque page, j'ai fait une condition if qui vérifie l'utilisateur connecte et affiche la page si l'utilisateur a le droit de voir la page (élèves ont le droit d'accéder à l'ajout et a la liste des entreprises et des secteurs et les professeurs ont le droit d'accès a toutes les pages).

```
const [ activeUser, setActiveUser ] = useState('x')  
  
if (activeUser === 'ancienselevesprof.stagnes@gmail.com') {  
  return (  

```

7. Page de contact

J'ai aussi créé une page de contact permettant de récupérer les adresses mails des élèves et de remplir un formulaire pour envoyer un mail :

Mails des anciens élèves :

celineduhil1983@gmail.com,fbolsard@sna-vernon27.fr,charlene.P74@hotmail.fr,guillaume.dorothe@hotmail.com,kathy-alexandre@wanadoo.fr,enzo_lola.ak@gmail.com,

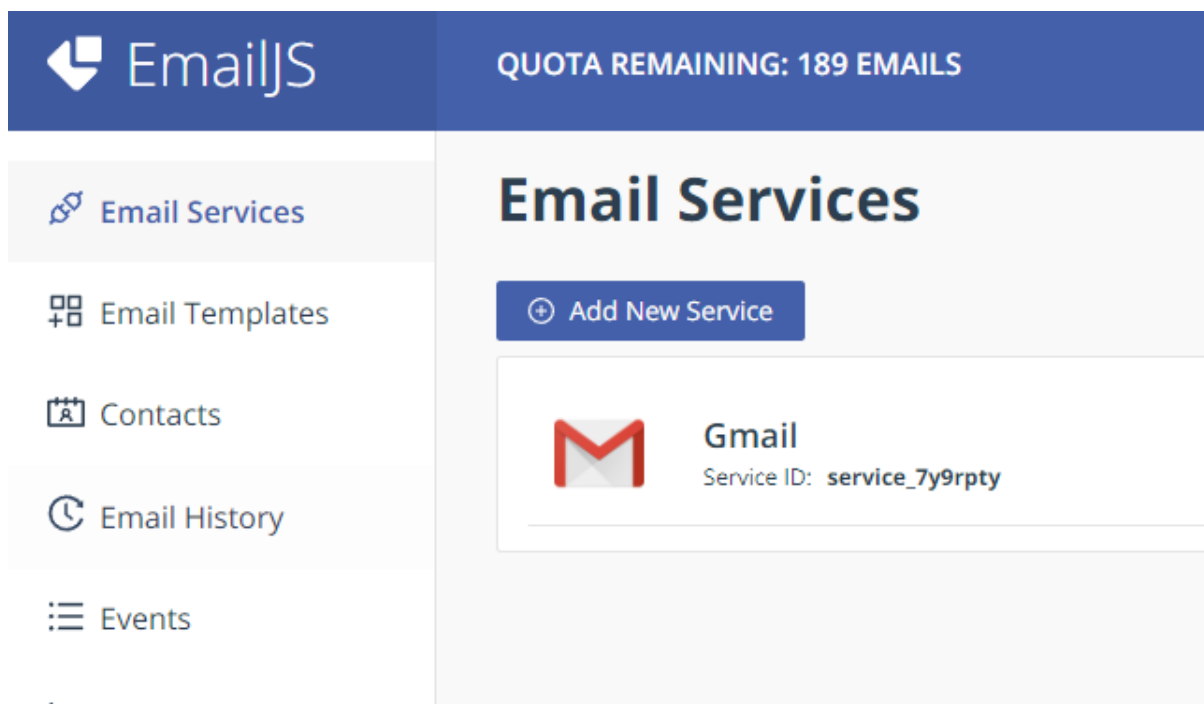
Envoyer un mail :

Destinataires :

Sujet :

Message :

Pour gérer l'envoi de mails j'ai utilisé le service EmailJS.



8. Documentation

J'ai fait en sorte de rendre mon code compréhensible et de le commenter régulièrement et j'ai transmis les codes d'accès au mail et au site a la personne responsable du projet.

CONCLUSION

1. Ce que j'ai appris

Ce stage m'a permis d'approfondir et de consolider mes connaissances en Javascript et en React et de maîtriser les Api et les envois de paramètres via une interface graphique.

J'ai aussi découvert et utilisé des services qui me seront utiles dans mes futurs projets comme EmailJS et Firebase.

2. Remerciements

Je tiens à remercier le lycée Saint Agnès de m'avoir accepté en stage au sein de leur établissement et de m'avoir permis d'avoir une nouvelle expérience professionnelle.

Merci à Mme Schmitzler de m'avoir intégré dans son projet et dans sa classe pendant 7 semaines et de m'avoir encadrée avec bienveillance et confiance.