



المدرسة العليا
للتكنولوجيا - الصويرة
L'ÉCOLE SUPÉRIEURE DE
TECHNOLOGIE – ESSAOUIRA

Micro Projet

Réalisation d'un site web pour l'achat de la musique avec
Spring Boot , Maven ,ReactJs ,Material-UI et MySQL ..

Hmimssa Soufiane

2020/2021

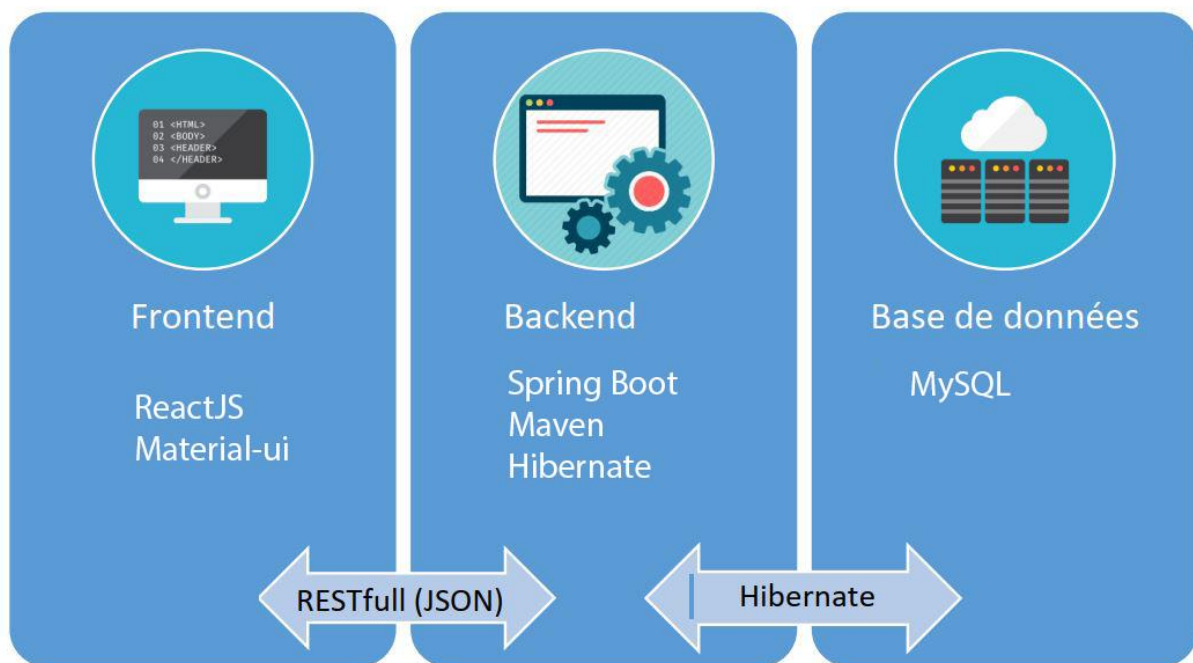
Objectif

Réalisation d'un site web pour l'achat de la musique. L'architecture utilisée tout au long du développement est divisée en trois parties.

L'architecture utilisée

L'architecture utilisée est l'architecture microservices qui divise une application comme un ensemble de petits services. Chaque service fonctionne moyennant son propre processus qui communique avec des mécanismes légers. Les services sont développés autour des compétences métiers qui sont déployés d'une façon indépendante par un processus automatisé. Ces services sont isolés et autonomes mais ils communiquent entre eux pour fournir les fonctionnalités nécessaires.

Cette architecture présente plusieurs avantages comme l'hétérogénéité technologique, la résistance contre l'échec, la scalabilité sur mesure, la facilité de déploiement, l'alignement organisationnel, la réutilisabilité, etc.



Les outils de développement et les technologies utilisés

Afin de réaliser cette application dynamique, je me suis servi des outils et langages suivants :

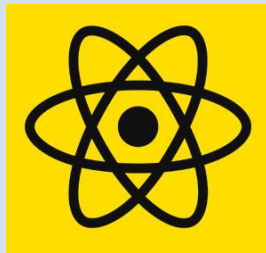
➤ Material UI :



➤ Spring Boot :



➤ ReactJS :



➤ MySQL :



➤ Hibernate :



➤ Maven:



➤ GitHub



➤ Xampp



➤ Eclipse



➤ Visual Studio Code



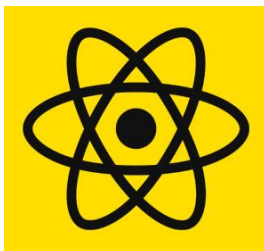
Description des technologies et Outils de développement



Spring est un framework qui facilite le développement d'applications fondées sur **Spring** en offrant des outils permettant d'obtenir une application packagée en jar .



MySQL est un système libre de gestion de base de données relationnelle (SGBDR) utilisant le langage SQL (Structured Query Language), soit le langage le plus populaire en matière d'ajout, d'accès et de traitement des données dans une base de données. Il est principalement reconnu pour sa rapidité, sa fiabilité et sa flexibilité



React est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page HTML à chaque changement d'état



HIBERNATE

Hibernate est un framework open source gérant la persistance des objets en base de données relationnelle



Material-UI C'est un ensemble des composants UI qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.



Apache Maven est un outil de gestion et d'automatisation de production des projets logiciels Java en général et Java EE en particulier. Il est utilisé pour automatiser l'intégration continue lors d'un développement de logiciel. Maven est géré par l'organisation Apache Software Foundation.



XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide.



Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.



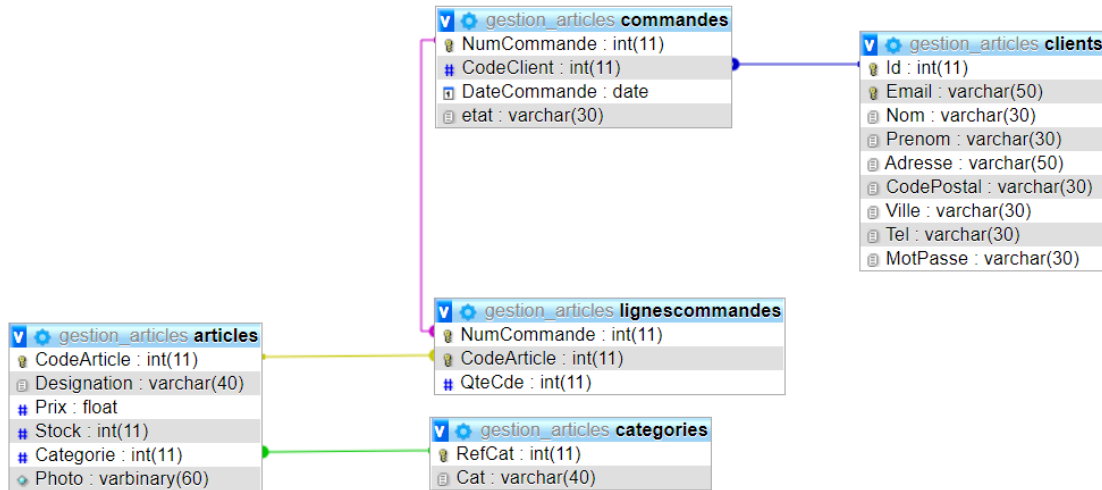
Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.



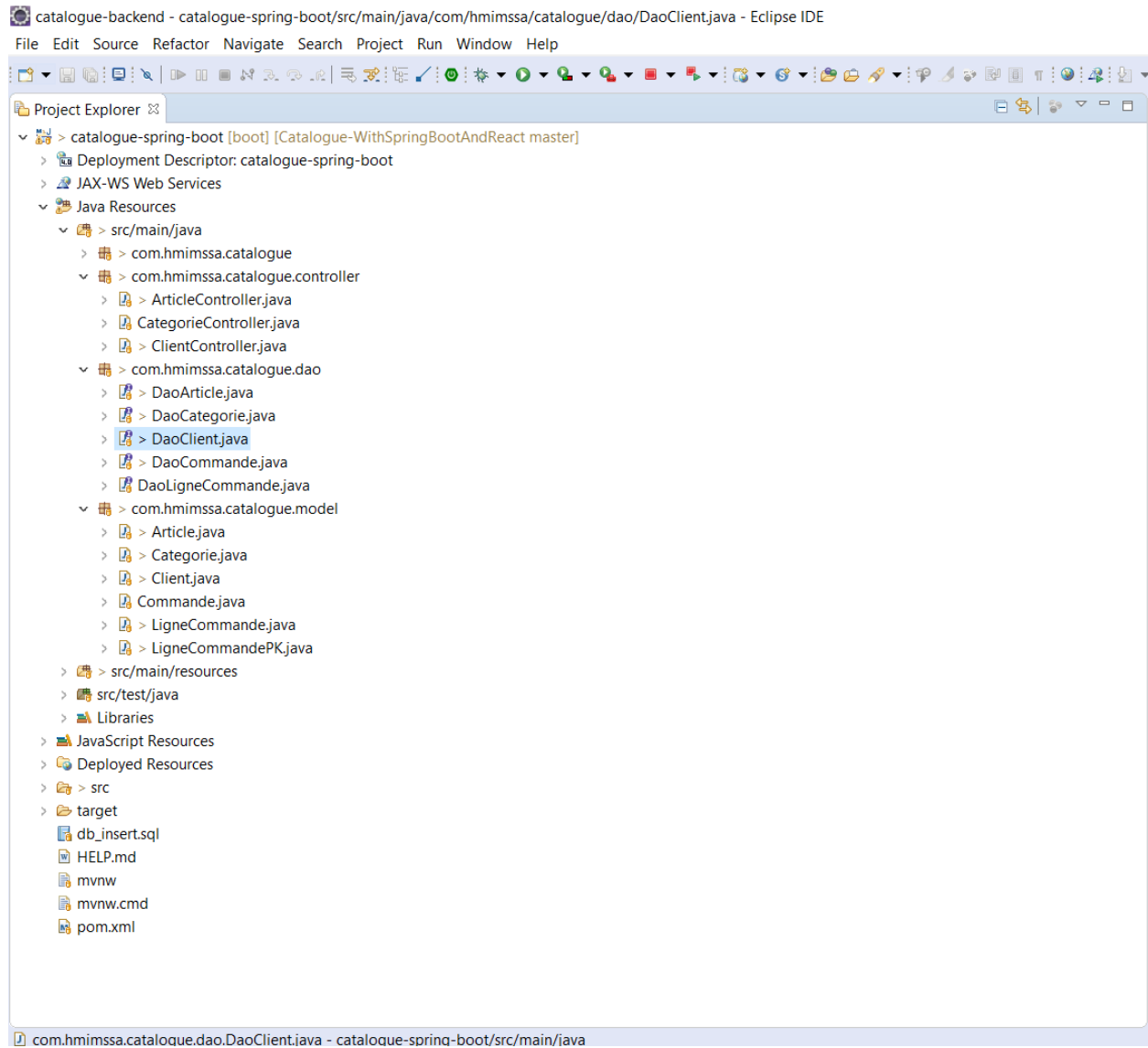
GitHub exploité sous le nom de GitHub, Inc. est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Ce site est développé en Ruby on Rails et Erlang par Chris Wanstrath.

Notre base de données

Cette base de données a été générée par **Hibernate**

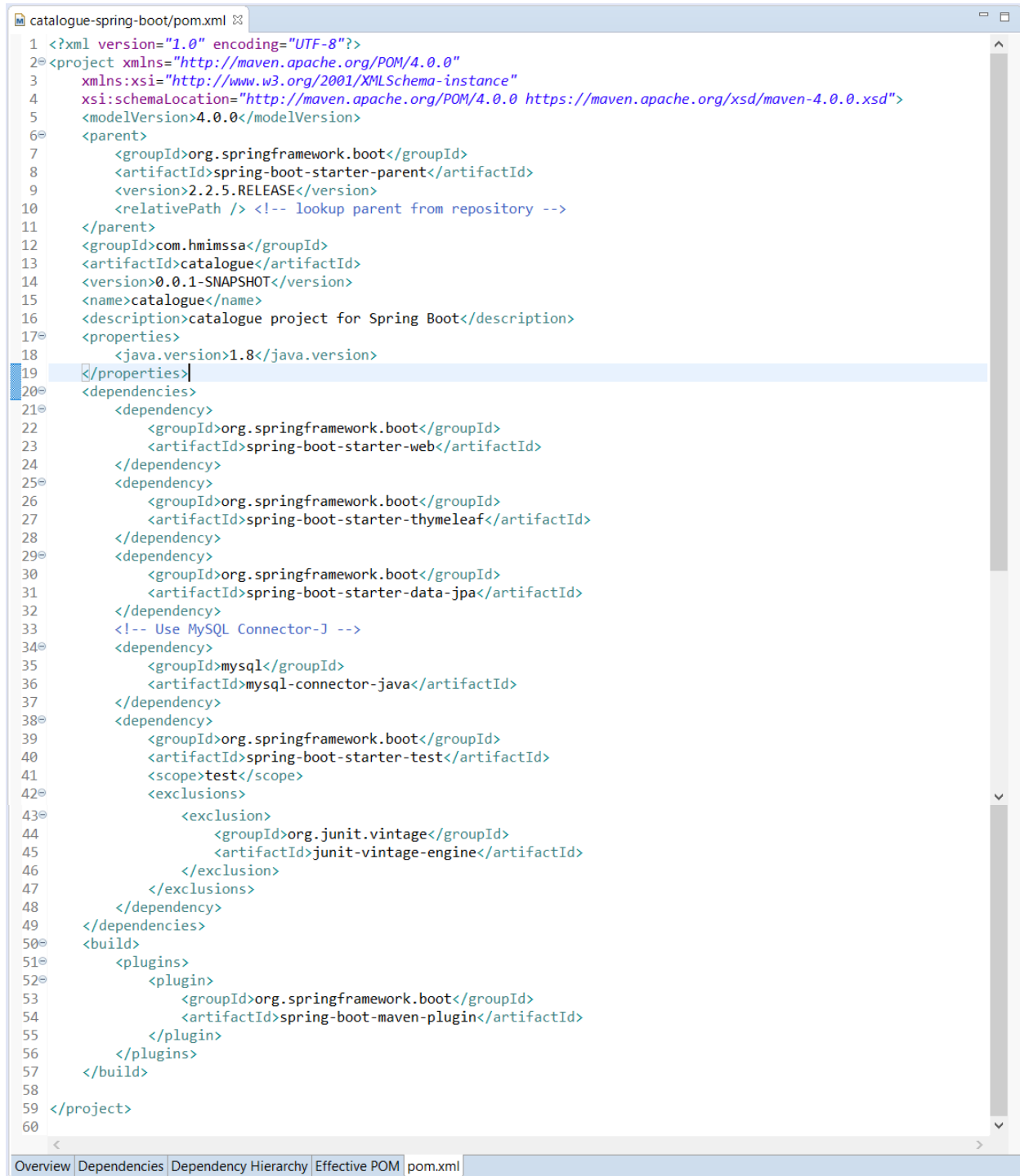


Structure de Projet Back-end



com.hmimssa.catalogue.dao.DaoClient.java - catalogue-spring-boot/src/main/java

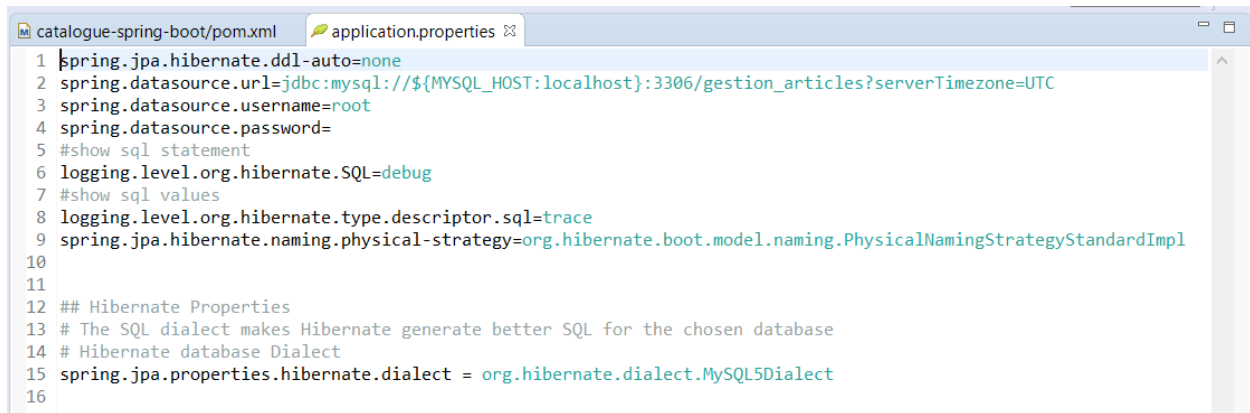
Configuration du fichier POM.XML



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6     <parent>
7         <groupId>org.springframework.boot</groupId>
8         <artifactId>spring-boot-starter-parent</artifactId>
9         <version>2.2.5.RELEASE</version>
10        <relativePath /> <!-- lookup parent from repository -->
11    </parent>
12    <groupId>com.hmimssa</groupId>
13    <artifactId>catalogue</artifactId>
14    <version>0.0.1-SNAPSHOT</version>
15    <name>catalogue</name>
16    <description>catalogue project for Spring Boot</description>
17    <properties>
18        <java.version>1.8</java.version>
19    </properties>
20    <dependencies>
21        <dependency>
22            <groupId>org.springframework.boot</groupId>
23            <artifactId>spring-boot-starter-web</artifactId>
24        </dependency>
25        <dependency>
26            <groupId>org.springframework.boot</groupId>
27            <artifactId>spring-boot-starter-thymeleaf</artifactId>
28        </dependency>
29        <dependency>
30            <groupId>org.springframework.boot</groupId>
31            <artifactId>spring-boot-starter-data-jpa</artifactId>
32        </dependency>
33        <!-- Use MySQL Connector-J -->
34        <dependency>
35            <groupId>mysql</groupId>
36            <artifactId>mysql-connector-java</artifactId>
37        </dependency>
38        <dependency>
39            <groupId>org.springframework.boot</groupId>
40            <artifactId>spring-boot-starter-test</artifactId>
41            <scope>test</scope>
42            <exclusions>
43                <exclusion>
44                    <groupId>org.junit.vintage</groupId>
45                    <artifactId>junit-vintage-engine</artifactId>
46                </exclusion>
47            </exclusions>
48        </dependency>
49    </dependencies>
50    <build>
51        <plugins>
52            <plugin>
53                <groupId>org.springframework.boot</groupId>
54                <artifactId>spring-boot-maven-plugin</artifactId>
55            </plugin>
56        </plugins>
57    </build>
58 </project>
```

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Configuration du fichier properties

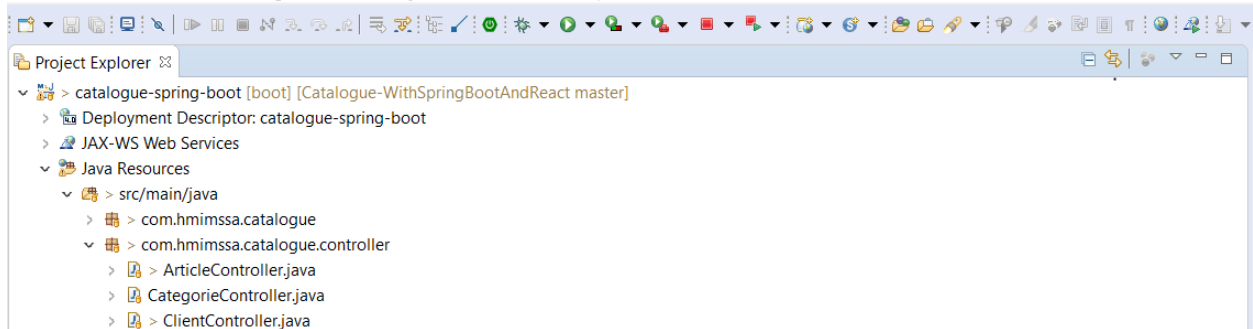


```
1 spring.jpa.hibernate.ddl-auto=none
2 spring.datasource.url=jdbc:mysql://${MYSQL_HOST:localhost}:3306/gestion_articles?serverTimezone=UTC
3 spring.datasource.username=root
4 spring.datasource.password=
5 #show sql statement
6 logging.level.org.hibernate.SQL=debug
7 #show sql values
8 logging.level.org.hibernate.type.descriptor.sql=trace
9 spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
10
11
12 ## Hibernate Properties
13 # The SQL dialect makes Hibernate generate better SQL for the chosen database
14 # Hibernate database Dialect
15 spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect
16
```

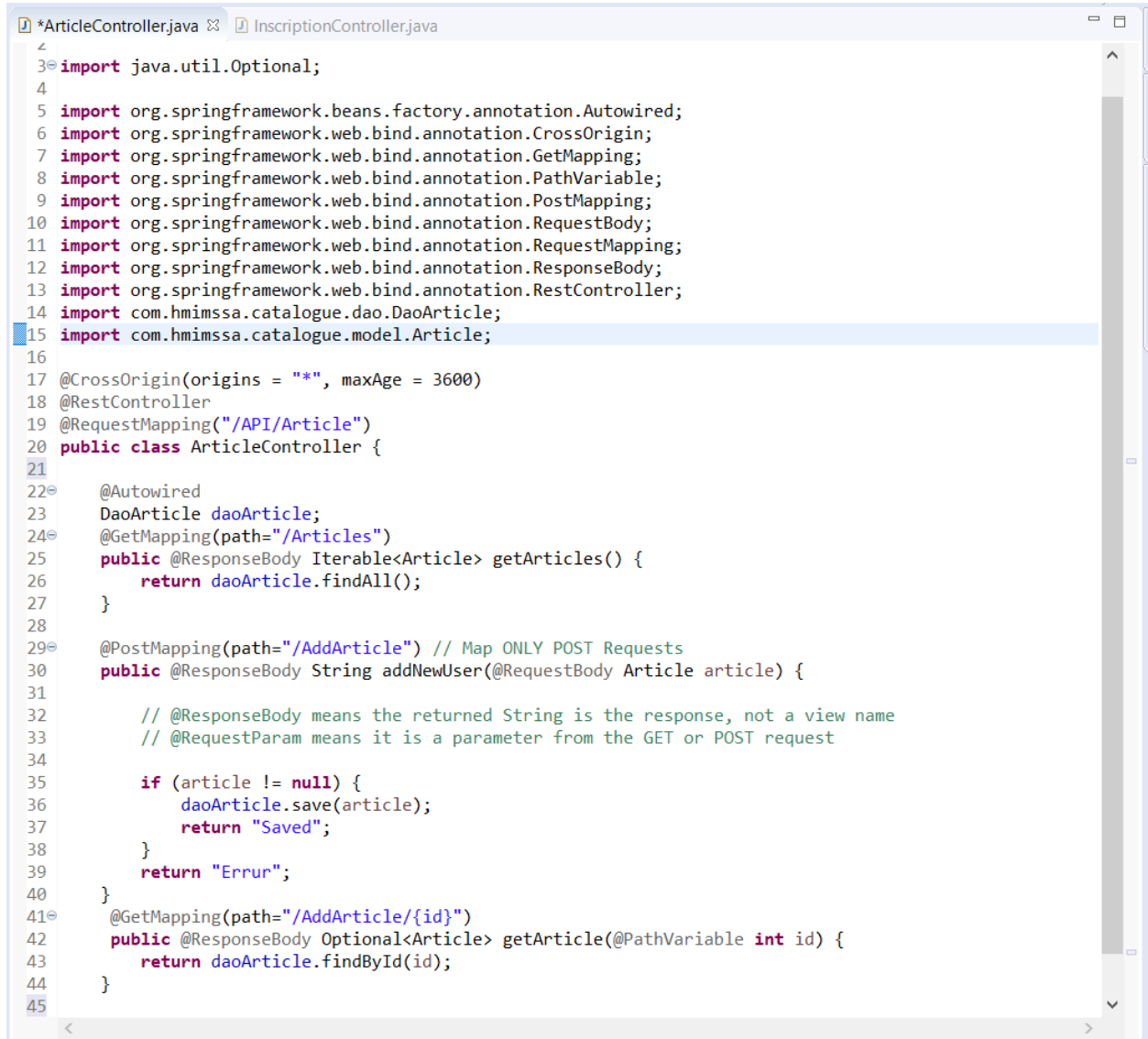
Couche Controller

catalogue-backend - catalogue-spring-boot/src/main/java/com/hmimssa/catalogue/dao/DaoClient.java - Eclipse IDE

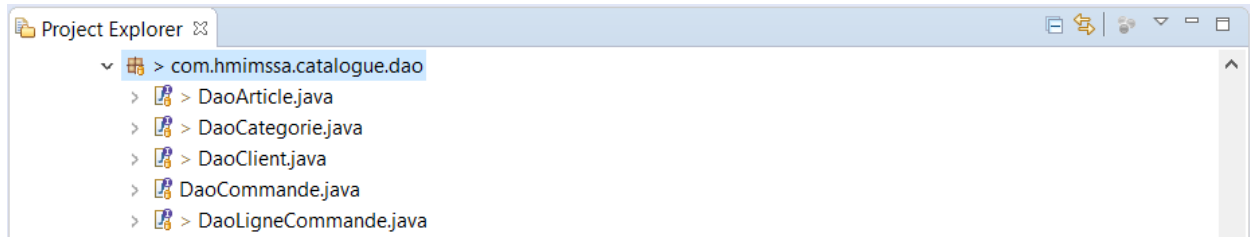
File Edit Source Refactor Navigate Search Project Run Window Help



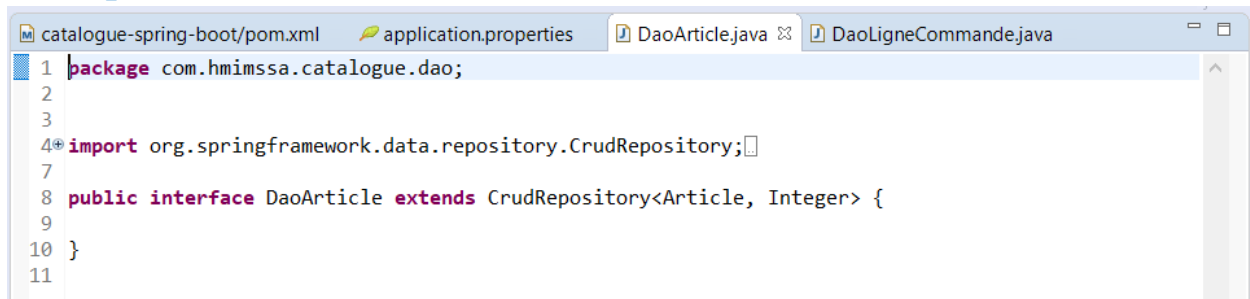
Exemple contrôleur d'Article :



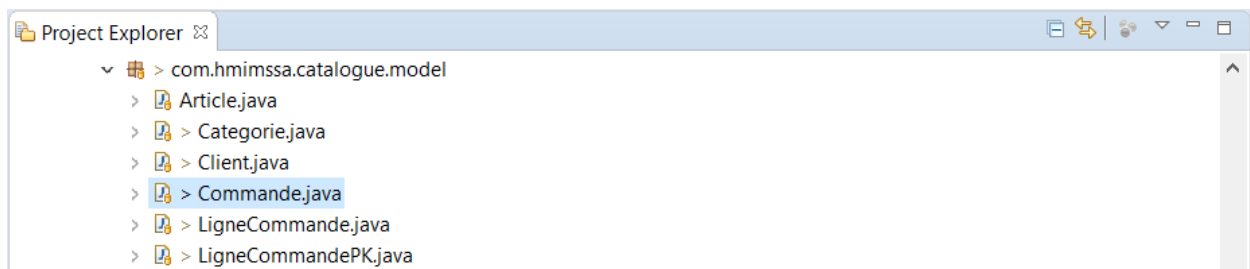
Couche Dao



Exemple :



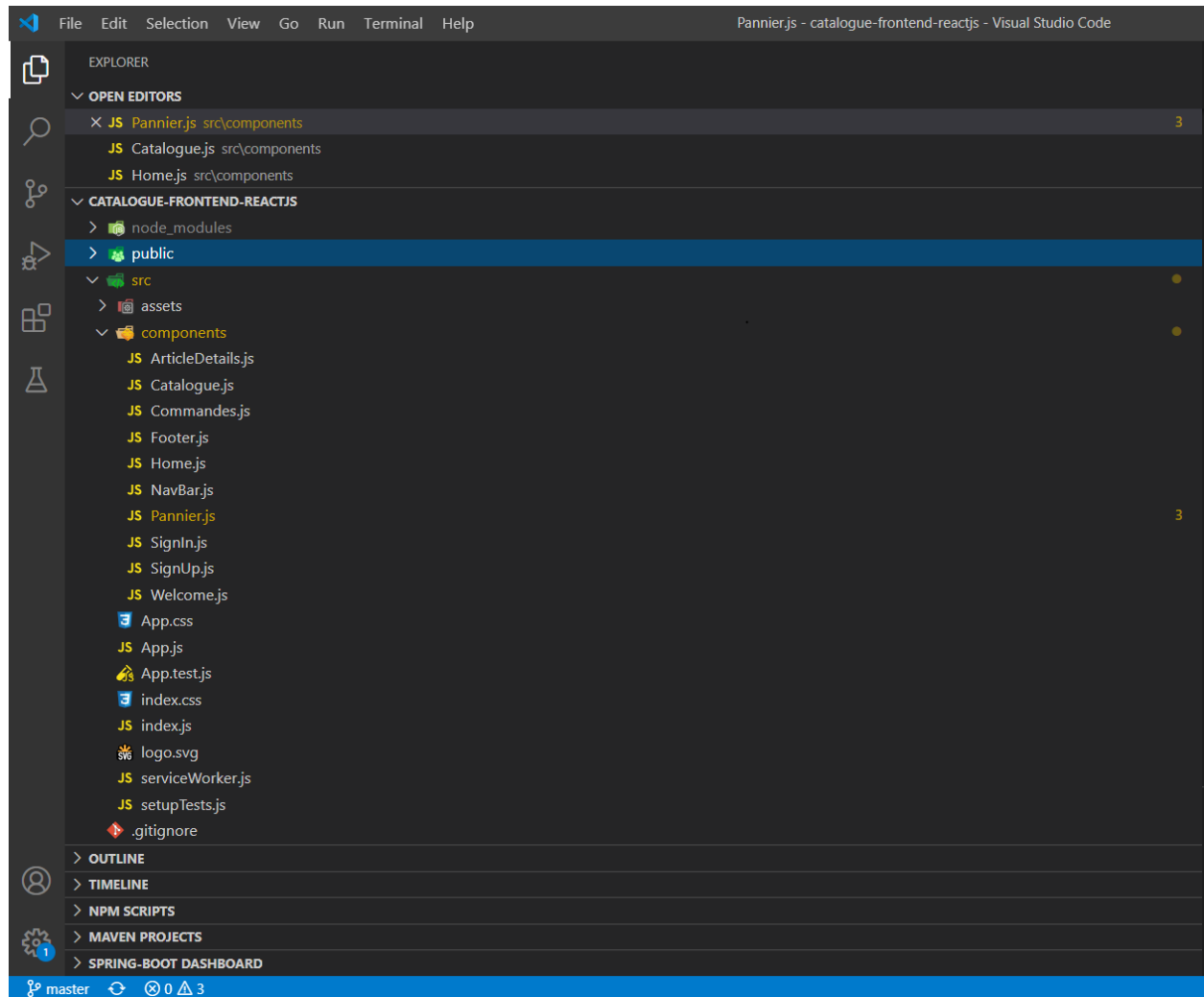
Couche Métier



Exemple de class commande de couche Metier

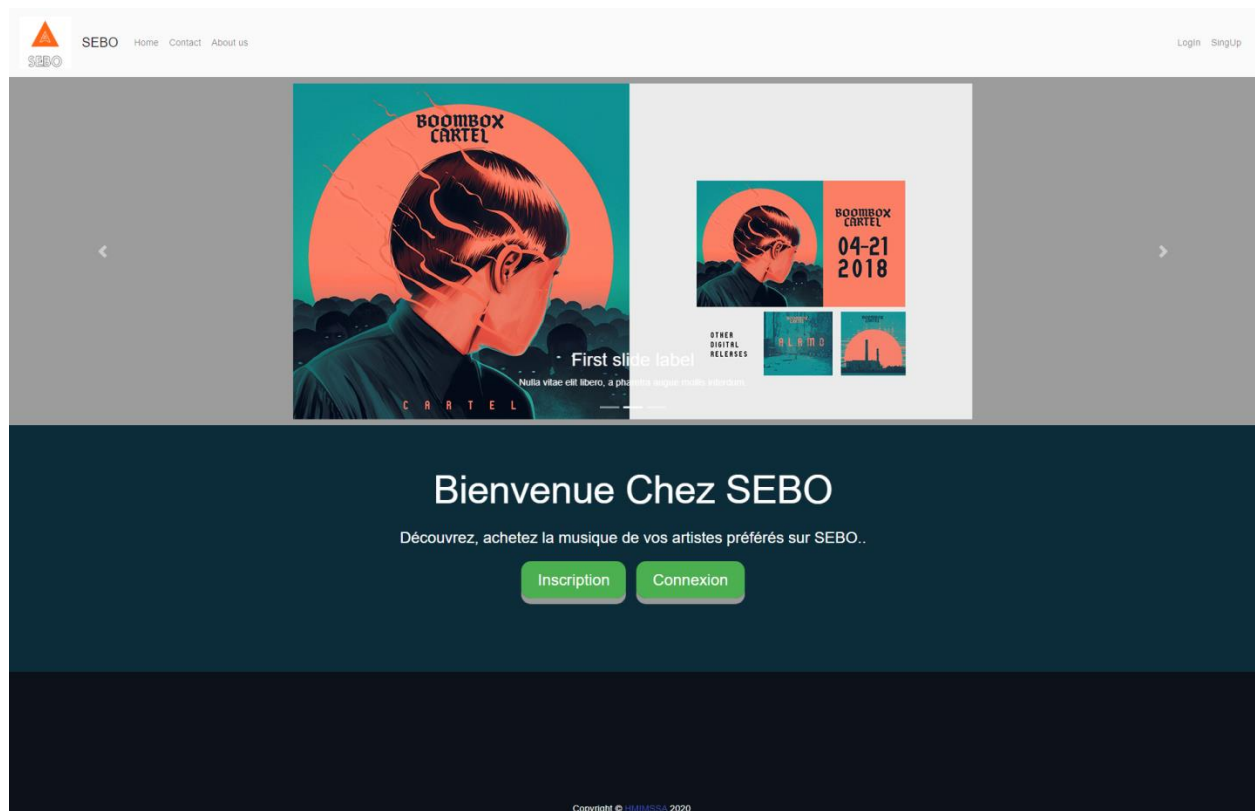
```
catalogue-spring-b... application.proper... DaoArticle.java DaoLigneCommande.java Commande.java
1 package com.hmimssa.catalogue.model;
2
3 import java.util.Date;
19
20 @Entity
21 public class Commande {
22     @Id
23     @GeneratedValue(strategy = GenerationType.IDENTITY)
24     private int numCommande;
25     private Date dateCommande;
26     private String etat;
27
28     @ManyToOne
29     @Cascade(value = { CascadeType.SAVE_UPDATE })
30     @JoinColumn(name="codeClient")
31     private Client client;
32     @OneToMany(mappedBy = "commande")
33     @Cascade(value = { CascadeType.ALL })
34     @SortNatural
35     @MapKey(name = "article")
36     private SortedMap<Article, LigneCommande> detailsCommandes = new TreeMap<>();
37
38     public Commande() {
39         super();
40     }
41
42     public Commande(int numCommande, Client client, Date dateCommande, String etat) {
43         super();
44         this.numCommande = numCommande;
45         this.client = client;
46         this.dateCommande = dateCommande;
47         this.etat = etat;
48     }
49
50     public int getNumCommande() {
51         return numCommande;
52     }
53
54     public void setNumCommande(int numCommande) {
55         this.numCommande = numCommande;
56     }
57
58     public Client getClient() {
59         return client;
60     }
61
62     public void setClient(Client client) {
63         this.client = client;
64     }
65
66     public Date getDateCommande() {
67         return dateCommande;
68     }
69
70     public void setDateCommande(Date dateCommande) {
71         this.dateCommande = dateCommande;
72     }
73
74     public String getEtat() {
75         return etat;
76     }
77
78     public void setEtat(String etat) {
79         this.etat = etat;
80     }
81 }
82
83
```

Structure de Projet Front-end



Partie User Interface

➤ Page de présentation:



Code :

```
import React, { useState } from 'react';
import Carousel from 'react-bootstrap/Carousel';
import Grid from '@material-ui/core/Grid';
import Typography from '@material-ui/core/Typography';
import { makeStyles } from '@material-ui/core/styles';
import Container from '@material-ui/core/Container';
import { Link } from "react-router-dom";
import home_banner from '../assets/home_banner1.png'; // with import
import home_banner2 from '../assets/home_banner2.png'; // with import
import home_banner3 from '../assets/home_banner3.png'; // with import

const useStyles = makeStyles(theme => ({
  icon: {
    marginRight: theme.spacing(2),
  },
  heroContent: {
    backgroundColor: theme.palette.background.paper,
    padding: theme.spacing(8, 0, 6),
  },
  heroButtons: {
    marginTop: theme.spacing(4),
  },
  cardGrid: {
    padding: theme.spacing(8),
  },
  card: {
    height: '100%',
    display: 'flex',
    flexDirection: 'column',
  },
  cardMedia: {
    padding: theme.spacing(8, 0, 6),
  },
  cardContent: {
    flex: 1,
  },
  button: {
    padding: "15px 25px",
    fontSize: "24px",
    cursor: "pointer",
    textAlign: "center",
    outline: "none",
    color: "#ffff0",
    backgroundColor: "#4CAF50",
    border: "none",
    borderRadius: "15px",
    boxShadow: "0 9px #999"
  },
}));

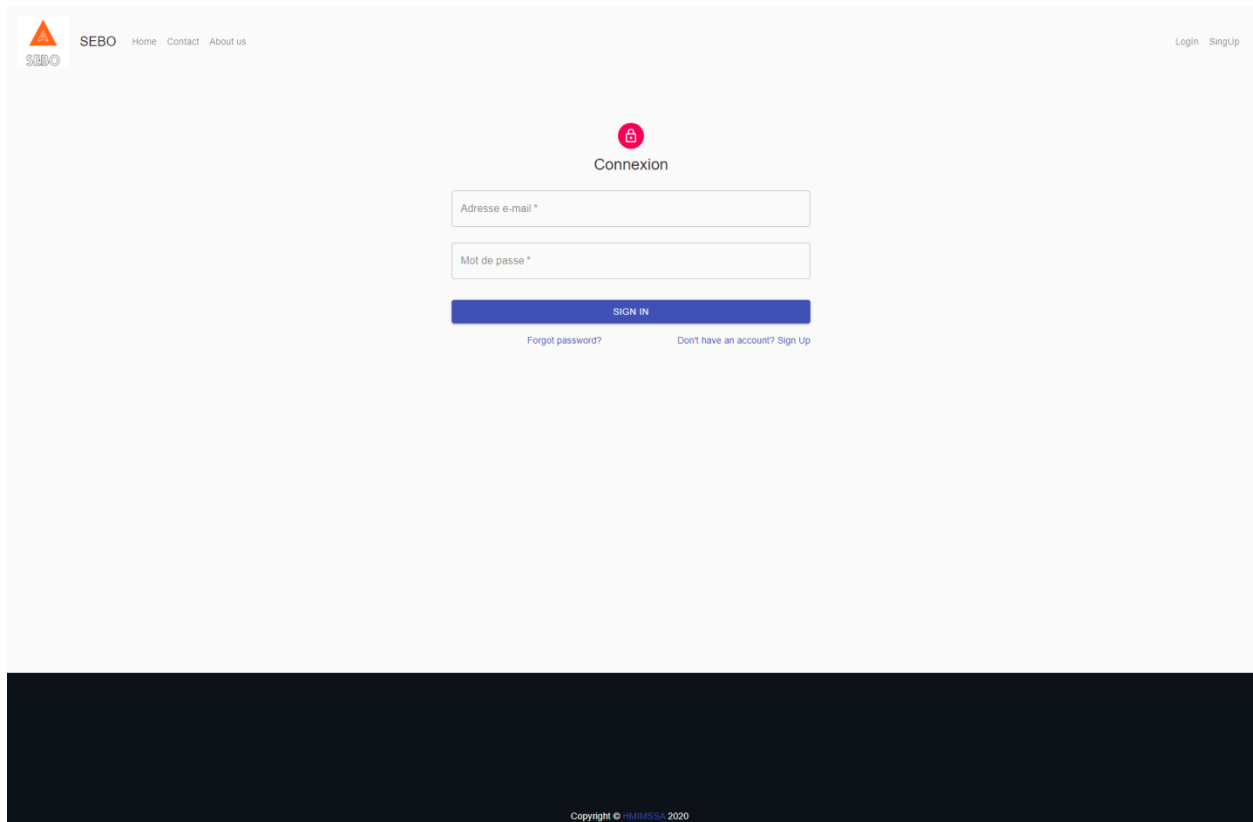
export default function Welcome() {
  const classes = useStyles();
  const [index, setIndex] = useState(0);

  const handleSelect = (selectedIndex, e) => {
    setIndex(selectedIndex);
  };

  return (
    <div style={{ height: "100vh", background: "#0c2c39" }} >
      <Carousel activeIndex={index} onSelect={handleSelect}>
        <Carousel.Item>
```


➤ Page d'authentification :

Dès l'accès à l'application une page d'authentification s'affiche.
Après authentification accéder aux différentes fonctionnalités de l'site



```
import React, { useState } from 'react';
import Avatar from '@material-ui/core/Avatar';
import Button from '@material-ui/core/Button';
import CssBaseline from '@material-ui/core/CssBaseline';
import TextField from '@material-ui/core/TextField';
import Link from '@material-ui/core/Link';
import Grid from '@material-ui/core/Grid';
import LockOutlinedIcon from '@material-ui/icons/LockOutlined';
import Typography from '@material-ui/core/Typography';
import { makeStyles } from '@material-ui/core/styles';
import Container from '@material-ui/core/Container';

const useStyles = makeStyles((theme) => ({
  paper: {
    marginTop: theme.spacing(8),
    display: 'flex',
    flexDirection: 'column',
    alignItems: 'center',
  },
  avatar: {
    margin: theme.spacing(1),
    backgroundColor: theme.palette.secondary.main,
  },
  form: {
    width: '100%', // Fix IE 11 issue.
  },
}));
```

```

        marginTop: theme.spacing(1),
      },
      submit: {
        margin: theme.spacing(3, 0, 2),
      },
    }));
  }

export default function SignIn(props) {
  const classes = useStyles();
  const [client, setClient] = useState({});

  const [email, setEmail] = useState("");
  const [motPasse, setmotPasse] = useState("");
  const [redirect, setRedirect] = useState(false);

  const renderRedirect = () => {
    if (redirect) {
      // return <Redirect to='/Home' />
      props.history.push({
        pathname: "/Home",
        state: {
          o: client
        }
      });
    }
  }

  const onSubmit = (e) => {
    e.preventDefault();

    var myHeaders = new Headers();
    myHeaders.append("Content-Type", "application/json");

    var raw = JSON.stringify({ "email": email, "motPasse": motPasse });

    var requestOptions = {
      method: 'POST',
      headers: myHeaders,
      body: raw,
      redirect: 'follow'
    };

    fetch("http://localhost:8080/API/Client/signIn", requestOptions)
      .then(response => response.text())
      .then(result => {
        if (result !== "") { setClient(JSON.parse(result)); setRedirect(true); }
      })
      .catch(error => console.log('error', error));

  }

  return (
    <div>
      {renderRedirect()}
      <Container style={{ height: "100vh" }} component="main" maxWidth="sm">
        <CssBaseline />
        <div className={classes.paper}>
          <Avatar className={classes.avatar}>
            <LockOutlinedIcon />
          </Avatar>
          <Typography component="h1" variant="h5">
            Connexion
          </Typography>
          <form className={classes.form} onSubmit={onSubmit} noValidate>
            <TextField variant="outlined" margin="normal" required fullWidth
              id="email"
              label="Adresse e-mail"
              name="email"
              onChange={e => {
                setEmail(e.target.value);
              }}
            />
          </form>
        </div>
      </Container>
    </div>
  );
}

```

```

        autoFocus
      />
      <TextField variant="outlined" margin="normal" required fullWidth
        name="password"
        label="Mot de passe"
        type="password"
        id="password"
        onChange={e => {
          setmotPasse(e.target.value);
        }}
        autoComplete="current-password"
      />
      <Button type="submit" fullWidth variant="contained" color="primary"
        className={classes.submit} >
        Sign In
      </Button>
      <Grid container>
        <Grid item xs>
          <Link href="#" variant="body2">
            Forgot password?
          </Link>
        </Grid>
        <Grid item>
          <Link href="#" variant="body2">
            {"Don't have an account? Sign Up"}
          </Link>
        </Grid>
      </Grid>
    </form>
  </div>

  </Container>
</div>
);
}

```

➤ Page Inscription:

The screenshot shows a web page for 'SEBO'. The header contains the logo, the name 'SEBO', and navigation links: 'Home', 'Contact', 'About us'. On the right, there are links for 'Login' and 'SingUp'. The main content area is titled 'Inscription' with a red lock icon. Below the title is a registration form with four input fields: 'Prenom *', 'Nom *', 'Email *', and 'Mot de passe *'. A blue button labeled 'INSCRIPTION' is at the bottom of the form. The footer of the page, on a dark background, reads 'Copyright © HMIMSSA 2020'.

Code :

```
import React, { useState } from 'react';
import Avatar from '@material-ui/core/Avatar';
import Button from '@material-ui/core/Button';
import CssBaseline from '@material-ui/core/CssBaseline';
import TextField from '@material-ui/core/TextField';
import Grid from '@material-ui/core/Grid';
import LockOutlinedIcon from '@material-ui/icons/LockOutlined';
import Typography from '@material-ui/core/Typography';
import { makeStyles } from '@material-ui/core/styles';
import Container from '@material-ui/core/Container';

const useStyles = makeStyles(theme => ({
  paper: {
    marginTop: theme.spacing(8),
    display: 'flex',
    flexDirection: 'column',
    alignItems: 'center',
  },
  avatar: {
    margin: theme.spacing(1),
    backgroundColor: theme.palette.secondary.main,
  },
  form: {
    width: '100%', // Fix IE 11 issue.
    marginTop: theme.spacing(3),
  },
  submit: {
    margin: theme.spacing(3, 0, 2),
  },
}));
```

```

export default function SignUp() {
  const classes = useStyles();
  const [client, setClient] = useState({});
  const [nom, setNom] = useState("");
  const [prenom, setPrenom] = useState("");
  const [email, setEmail] = useState("");
  const [motPasse, setmotPasse] = useState("");
  const [compteType, setCompteType] = useState('');

  const handleChange = (event) => {
    setCompteType(event.target.value);
  };

  const handleSubmit = (evt) => {
    evt.preventDefault();

    //var raw = JSON.stringify({ "email": email, "password": password });

    var myHeaders = new Headers();
    myHeaders.append("Content-Type", "application/json");

    setClient({
      "nom": nom,
      "prenom": prenom,
      "email": email,
      "motPasse": motPasse
    })
    var raw = JSON.stringify(client);
    var requestOptions = {
      method: 'POST',
      headers: myHeaders,
      body: raw,
      redirect: 'follow'
    };

    fetch("http://localhost:8080/API/Client/signUp", requestOptions)
      .then(response => response.text())
      .then(result => console.log(result))
      .catch(error => console.log('error', error));

  }

  return (
    <Container component="main" style={{ height: "100%" }} maxWidth="xs">
      <CssBaseline />
      <div className={classes.paper}>
        <Avatar className={classes.avatar}>
          <LockOutlinedIcon />
        </Avatar>
        <Typography component="h1" variant="h5">
          Inscription
        </Typography>
        <form className={classes.form} noValidate>
          <Grid container spacing={2}>
            <Grid item xs={12} sm={6}>
              <TextField autoComplete="fname" name="firstName"
                variant="outlined"
                required
                fullWidth
                id="firstName"
                label="Prenom"
                autoFocus
                onChange={e => {
                  setPrenom(e.target.value);
                }}
              />
            </Grid>

```

```

        <Grid item xs={12} sm={6}>
            <TextField variant="outlined" required fullWidth
                id="lastName"
                label="Nom"
                name="lastName"
                autoComplete="lname"
                onChange={e => {
                    setNom(e.target.value);
                }}
            />
        </Grid>
        <Grid item xs={12}>
            <TextField variant="outlined" required fullWidth
                id="email"
                label="Email"
                name="email"
                autoComplete="email"
                onChange={e => {
                    setEmail(e.target.value);
                }}
            />
        </Grid>
        <Grid item xs={12}>
            <TextField variant="outlined" required fullWidth
                name="password"
                label="Mot de passe"
                type="password"
                id="password"
                autoComplete="current-password"
                onChange={e => {
                    setmotPasse(e.target.value);
                }}
            />
        </Grid>

        </Grid>
        <Button type="submit" fullWidth variant="contained" color="primary"
            className={classes.submit}
        >
            Inscription
        </Button>

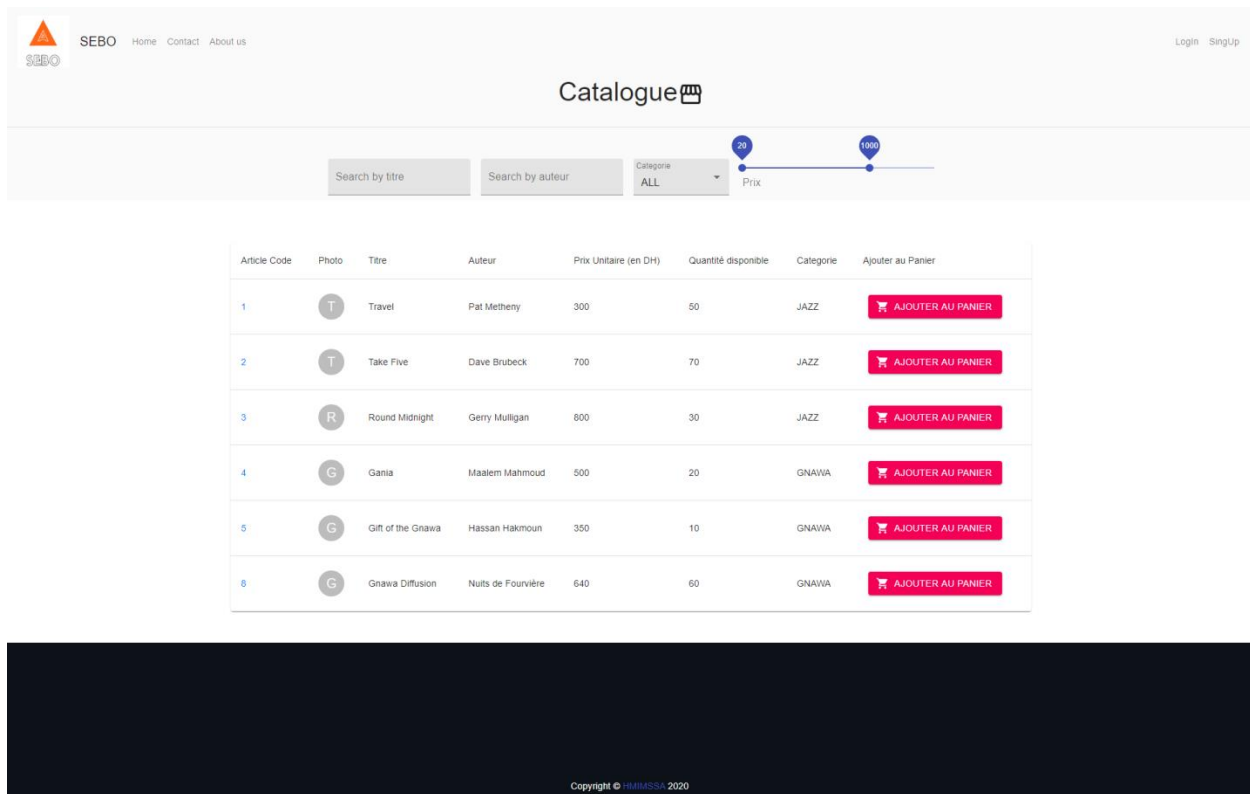
    </form>
</div>

</Container>
);
}

```

➤ Page Catalogue:

Filter par catégories, prix, titre, auteur.



Code :

```
import React, { useState, useEffect, useRef } from 'react';
import Table from '@material-ui/core/Table';
import TableBody from '@material-ui/core/TableBody';
import TableCell from '@material-ui/core/TableCell';
import TableContainer from '@material-ui/core/TableContainer';
import TableHead from '@material-ui/core/TableHead';
import TableRow from '@material-ui/core/TableRow';
import Paper from '@material-ui/core/Paper';
import Container from '@material-ui/core/Container';
import { withStyles } from '@material-ui/core/styles';
import TextField from '@material-ui/core/TextField';
import InputLabel from '@material-ui/core/InputLabel';
import FormControl from '@material-ui/core/FormControl';
import Select from '@material-ui/core/Select';
import Button from '@material-ui/core/Button';
import ShoppingCartIcon from '@material-ui/icons/ShoppingCart';
import Slider from '@material-ui/core/Slider';
import { Link } from 'react-router-dom';
import Avatar from '@material-ui/core/Avatar';
import StorefrontIcon from '@material-ui/icons/Storefront';

const styles = theme => ({
  icon: {
    marginRight: theme.spacing(2),
  },
  button: {
    margin: theme.spacing(1),
  },
  heroContent: {
```



```

        backgroundColor: theme.palette.background.paper,
        padding: theme.spacing(8, 0, 6),
    },
    margin: {
        margin: theme.spacing(1),
    },
    table: {
        minWidth: 650,
    },
    footer: {
        backgroundColor: theme.palette.background.paper,
        padding: theme.spacing(6),
    },
});

function valuetext(value) {
    return `${value}DH`;
}

function Catalogue(props) {
    const { classes } = props;
    const categoriesSelected = useRef(-1);
    const [articles, setArticles] = useState([]);
    const [articlesCopy, setArticlesCopy] = useState([]);
    const [filterByTitre, setFilterByTitre] = useState("");
    const [filterByAuteur, setFilterByAuteur] = useState("");
    const [categories, setCategories] = useState([]);
    const [value, setValue] = React.useState([20, 1000]);

    const handleChangeSlider = (event, newValue) => {
        setValue(newValue);
    };

    const handleChange = (event) => {
        categoriesSelected.current = parseInt(event.target.value);
        (categoriesSelected.current === -1) ? setArticlesCopy(articles) :
            setArticlesCopy(articles.filter(x => x.categorie.refCat === categoriesSelected.current));
    };

    const handleClick = (event) => {
        var requestOptions = {
            method: 'GET',
            redirect: 'follow'
        };

        fetch("http://localhost:8080/API/Article/articles", requestOptions)
            .then(response => response.text())
            .then(result => { setArticles(JSON.parse(result)); setArticlesCopy(JSON.parse(result)); })
            .catch(error => console.log('error', error));
    };

    useEffect(() => {
        var requestOptions = {
            method: 'GET',
            redirect: 'follow'
        };

        fetch("http://localhost:8080/API/Article/articles", requestOptions)
            .then(response => response.text())
            .then(result => { setArticles(JSON.parse(result)); setArticlesCopy(JSON.parse(result)); })
            .catch(error => console.log('error', error));
    }, []);

    useEffect(() => {
        var requestOptions = {
            method: 'GET',

```

```

    redirect: 'follow'
  });

  fetch("http://localhost:8080/API/Categorie/categories", requestOptions)
    .then(response => response.text())
    // .then(result => console.log(result))
    .then(result => setCategories(JSON.parse(result)))
    .catch(error => console.log('error', error));

}, []);

return (
  <main style={{ minHeight: "100%" }}>
    <div>
      <div style={{ marginBottom: "30px" }}>
        <h1><span style={{ verticalAlign: "bottom" }}>Catalogue</span>
          <StoreFrontIcon fontSize="inherit" />
        </h1>
      </div>
      <hr />
      <div style={{ marginTop: "40px" }}>
        <FormControl className={classes.margin}>
          <TextField id="filled-basic" onChange={e => {
            setFilterByTitre(e.target.value);
          }} label="Search by titre" variant="filled" />
        </FormControl>
        <FormControl className={classes.margin}>
          <TextField id="filled-basic" onChange={e => {
            setFilterByAuteur(e.target.value);
          }} label="Search by auteur" variant="filled" />
        </FormControl>
        <FormControl className={classes.margin}>
          <InputLabel style={{ padding: "5px" }}
            htmlFor="outlined-age-native-simple">
            Categorie
          </InputLabel>
          <Select
            variant="filled"
            native
            value={categoriesSelected.current}
            onChange={handleChange}
            inputProps={{
              name: 'Categorie',
              id: 'filled-age-native-simple',
            }}
            >
            <option aria-label="None" value={-1}>ALL</option>
            {categories?.map(row => (
              <option key={row.refCat} value={row.refCat}>
                {row.cat}
              </option>
            ))}
          </Select>
        </FormControl>
        <FormControl className={classes.margin} style={{ width: "300px" }}>
          <InputLabel style={{ padding: "5px" }}>Prix</InputLabel>
          <Slider
            value={value}
            onChange={handleChangeSlider}
            aria-labelledby="range-slider"
            valueLabelDisplay="on"
            max={1500}
            getAriaValueText={valueText}
          />
        </FormControl>
      </div>
    </div>
  </main>
);

```


➤ Page Pannier:



SEBO Home Contact About us

Login SingUp

Pannier🛒

Article Details : 1	T	Travel	300	1	SUPPRIMER
Article Details : 2	T	Take Five	700	2	SUPPRIMER
Article Details : 3	R	Round Midnight	800	2	SUPPRIMER
Article Details : 4	G	Gania	500	2	SUPPRIMER
Tax			7 %	161.00	
Total				2451.00	
<div>Commander</div> <div>CONFIRMER LA COMMANDE</div>					

Copyright © HMIMSSA 2020

Code :

```
import React, { useState, useEffect, useRef } from 'react';
import Table from '@material-ui/core/Table';
import TableBody from '@material-ui/core/TableBody';
import TableCell from '@material-ui/core/TableCell';
import TableContainer from '@material-ui/core/TableContainer';
import TableRow from '@material-ui/core/TableRow';
import Paper from '@material-ui/core/Paper';
import Container from '@material-ui/core/Container';
import { withStyles } from '@material-ui/core/styles';
import { Link } from 'react-router-dom';
import Avatar from '@material-ui/core/Avatar';
import HighlightOffIcon from '@material-ui/icons/HighlightOff';
import Button from '@material-ui/core/Button';
import ShoppingCartIcon from '@material-ui/icons/ShoppingCart';
import DoneOutlineIcon from '@material-ui/icons/DoneOutline';
import TableFooter from '@material-ui/core/TableFooter';

const styles = theme => ({
  icon: {
    marginRight: theme.spacing(2),
  },
  button: {
    margin: theme.spacing(1),
  },
  heroContent: {
    backgroundColor: theme.palette.background.paper,
```