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

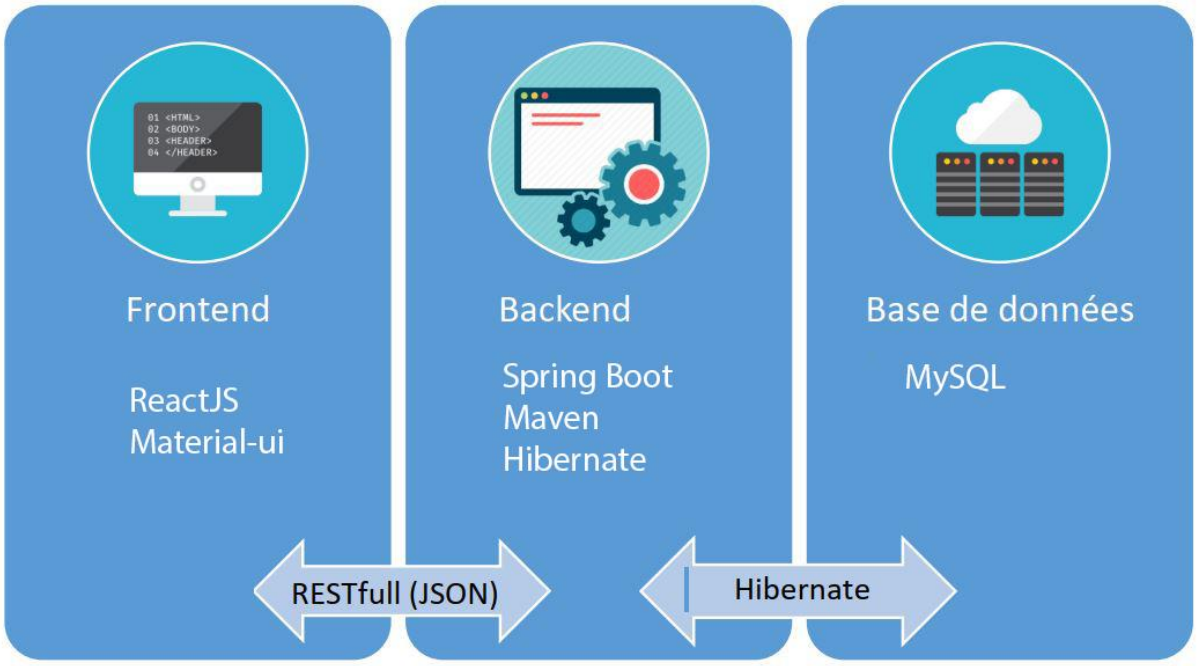
2020/2021

Objectif

Réalisation d'un site web pour l'achat de la musique avec l'architecture JEE

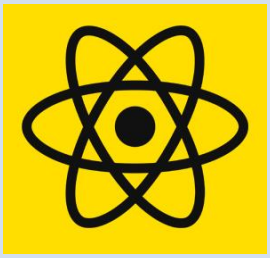
L'architecture utilisé

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 

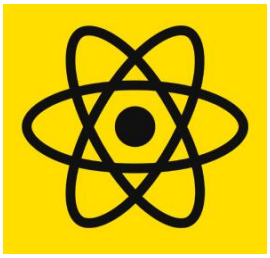
➢ Eclipse

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

Material-UI C'est un ensemble des components 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.

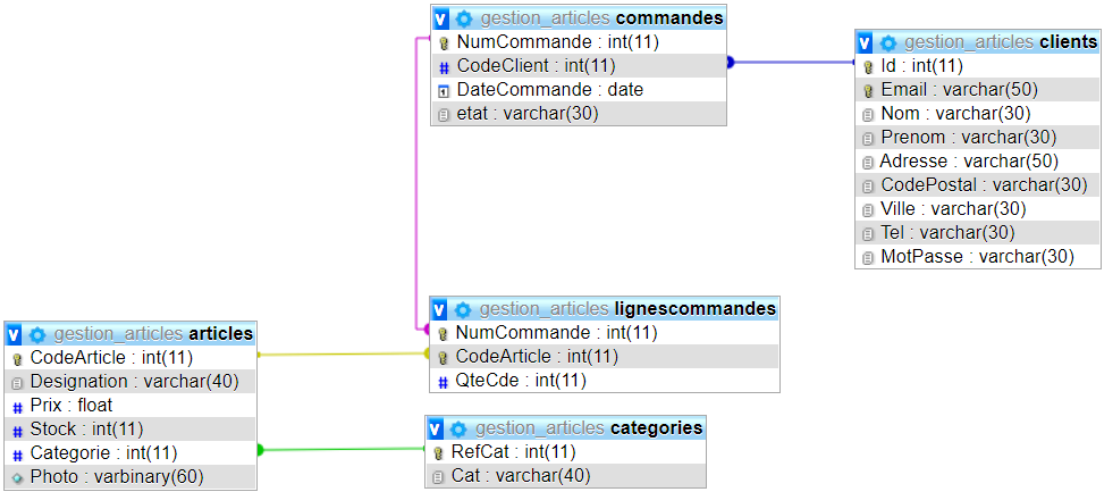
Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

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

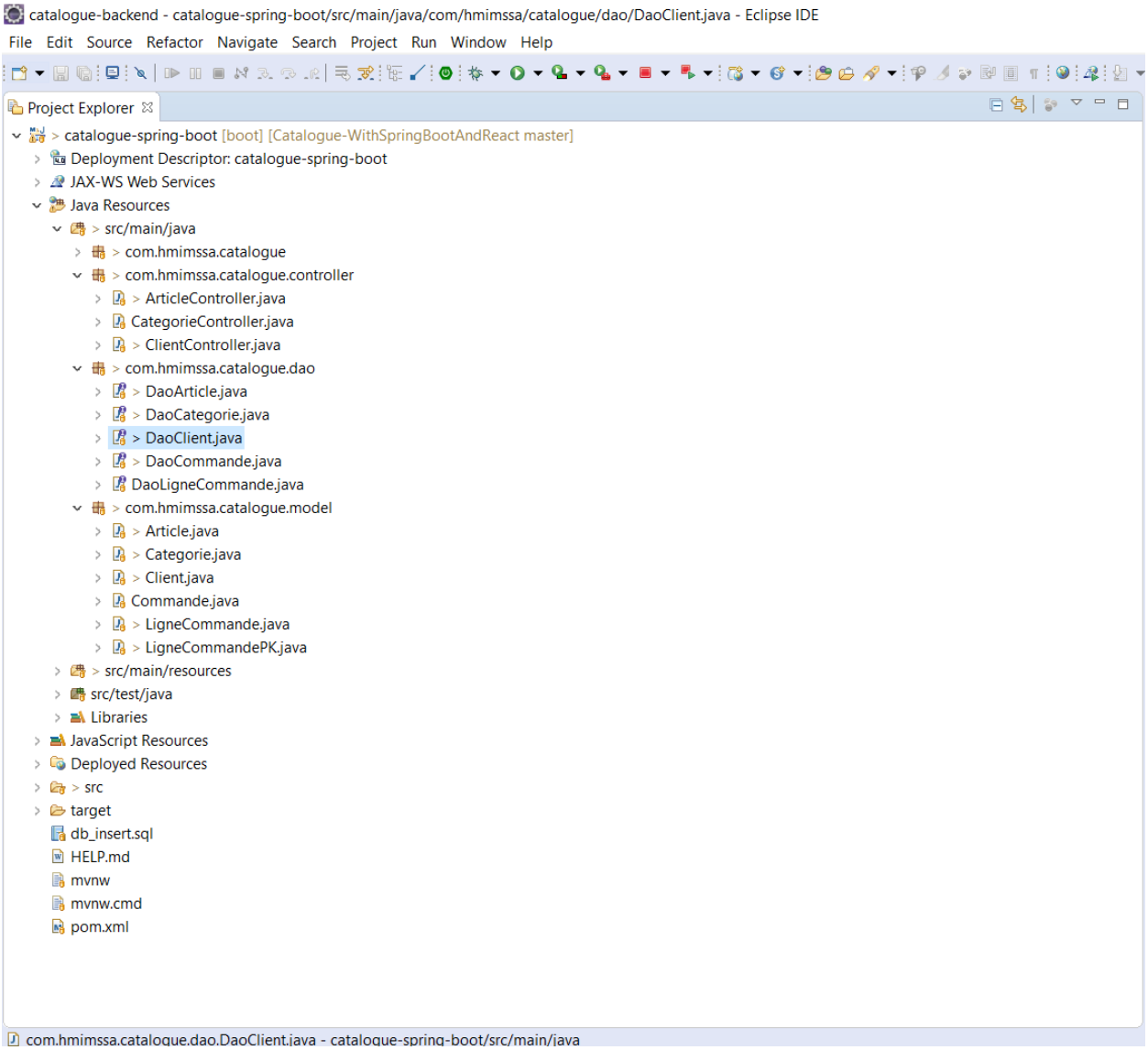
GitHub exploité sous le nom de GitHub, Inc. est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Ce site est développé en Ruby on Rails et Erlang par Chris Wanstrath.

Notre base de données

Cette base de donne a été générer par **Hibernate**



Structure de Projet Back-end



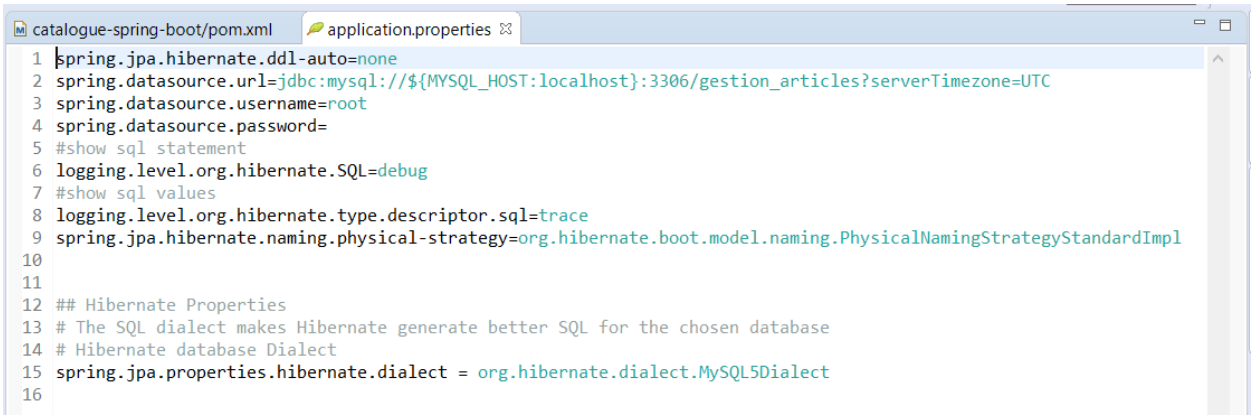
Configuration du fichier POM.XML

catalogue-spring-boot/pom.xml

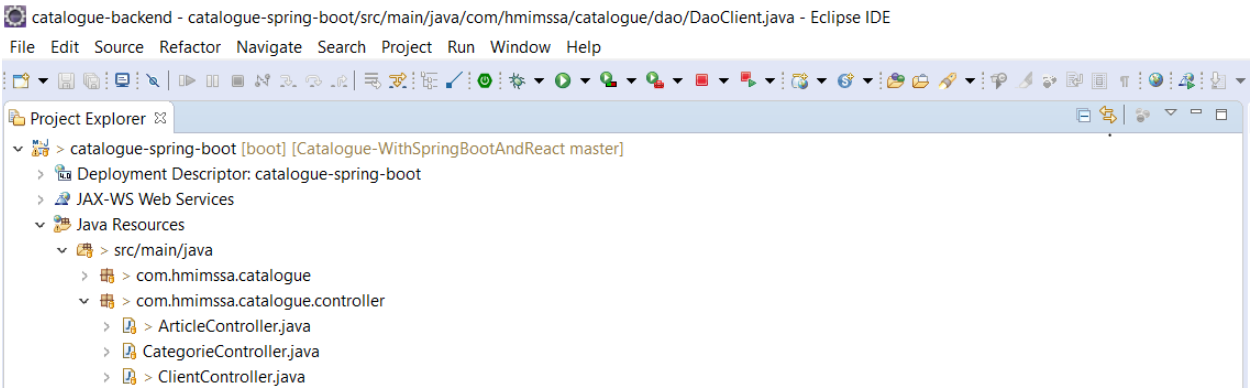
```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.2.5.RELEASE</version>
        <relativePath /> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.hmimssa</groupId>
    <artifactId>catalogue</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>catalogue</name>
    <description>catalogue project for Spring Boot</description>
    <properties>
        <java.version>1.8</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-thymeleaf</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <!-- Use MySQL Connector-J -->
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
            <exclusions>
                <exclusion>
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>

```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

Configuration du fichier properties



Couche Controller



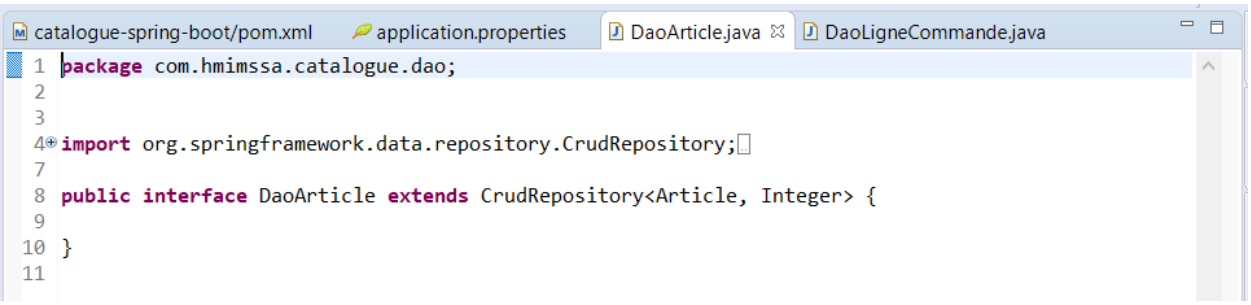
Exemple contrôleur d'Article :

*ArticleController.java InscriptionController.java

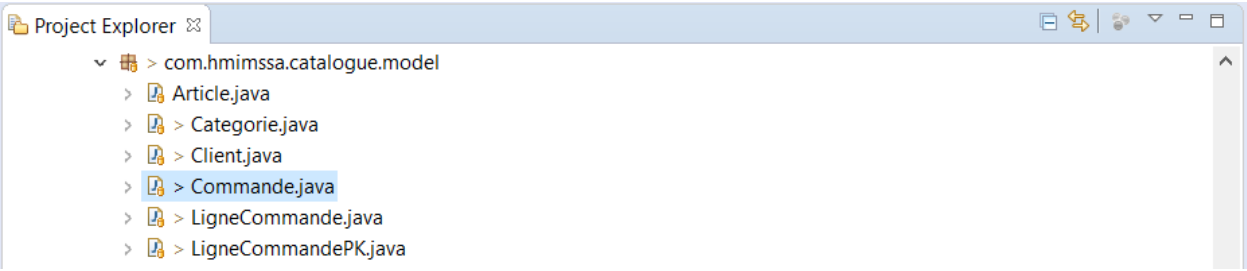
```java
import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.bind.annotation.RestController;
import com.hmimssa.catalogue.dao.DaoArticle;
import com.hmimssa.catalogue.model.Article;

@CrossOrigin(origins = "*", maxAge = 3600)
@RestController
@RequestMapping("/API/Article")
public class ArticleController {

    @Autowired
    DaoArticle daoArticle;
    @GetMapping(path="/Articles")
    public @ResponseBody Iterable<Article> getArticles() {
        return daoArticle.findAll();
    }

    @PostMapping(path="/AddArticle") // Map ONLY POST Requests
    public @ResponseBody String addNewUser(@RequestBody Article article) {

        // @ResponseBody means the returned String is the response, not a view name
        // @RequestParam means it is a parameter from the GET or POST request

        if (article != null) {
            daoArticle.save(article);
            return "Saved";
        }
        return "Errur";
    }
     @GetMapping(path="/AddArticle/{id}")
     public @ResponseBody Optional<Article> getArticle(@PathVariable int id) {
        return daoArticle.findById(id);
    }

```

Couche Dao

Exemple :



Couche Métier

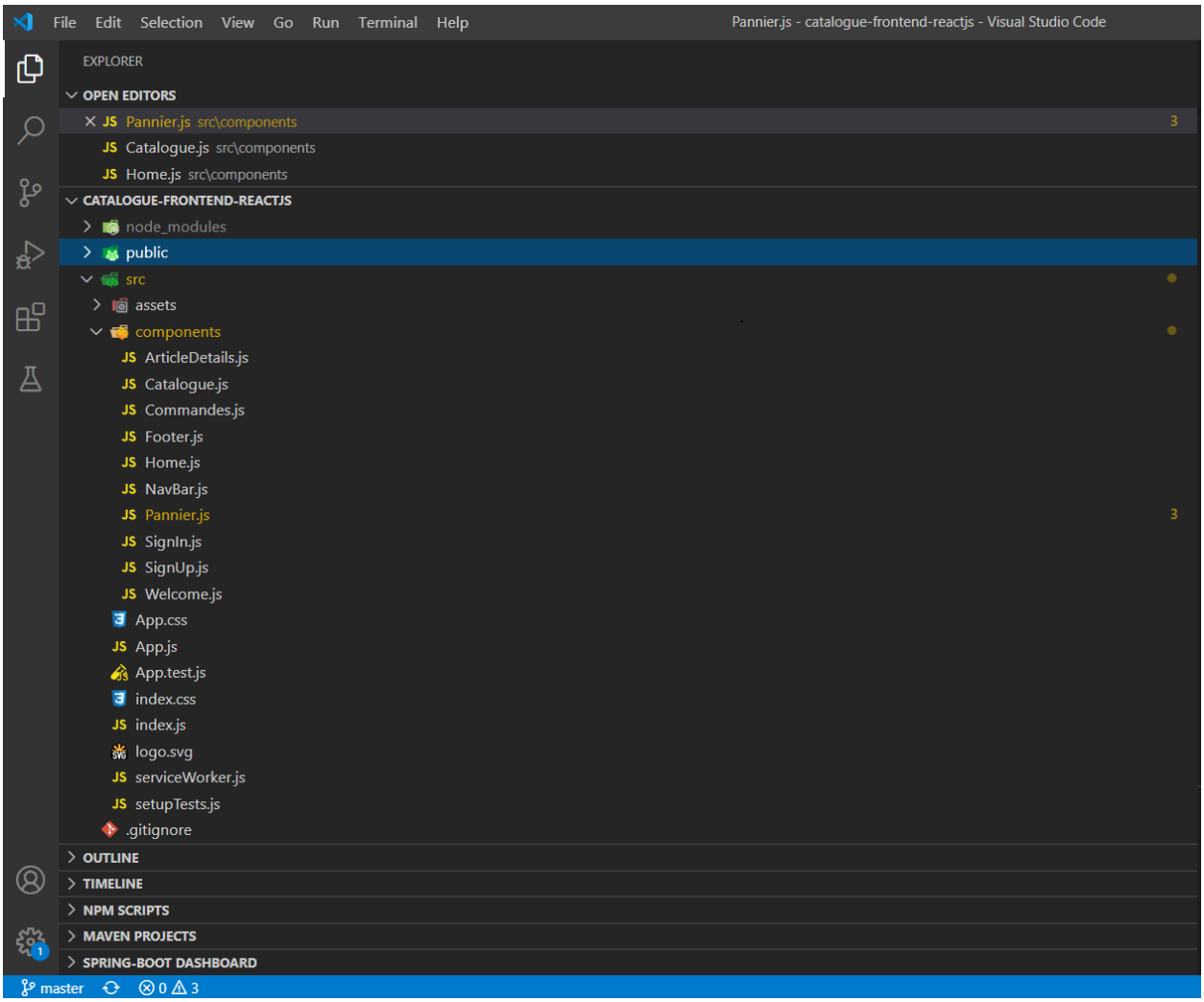


Exemple de class commande de couche Metier

catalogue-spring-b... | application.proper... | DaoArticle.java | DaoLigneCommande.j... | Commande.java

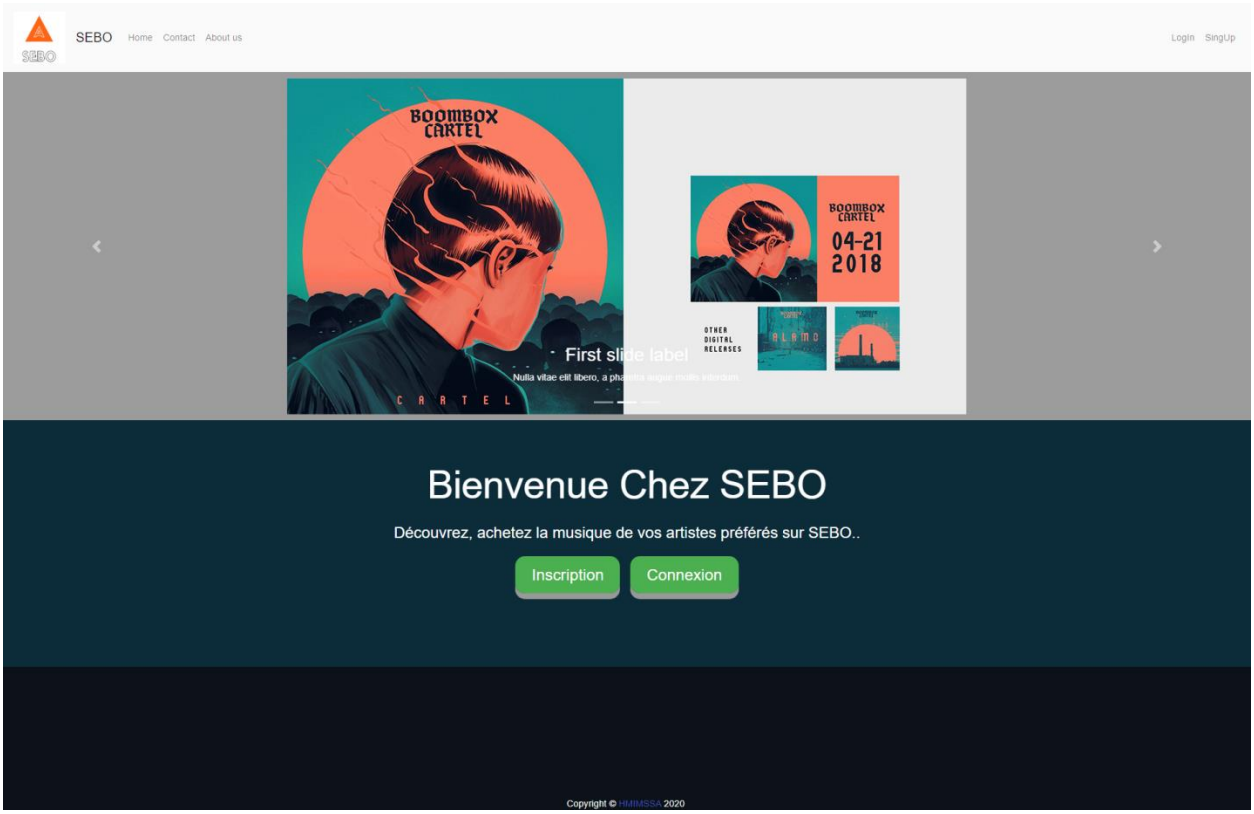
```java
package com.hmimssa.catalogue.model;

import java.util.Date;

@Entity
public class Commande {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int numCommande;
    private Date dateCommande;
    private String etat;

    @ManyToOne
    @Cascade(value = { CascadeType.SAVE_UPDATE })
    @JoinColumn(name="codeClient")
    private Client client;
    @OneToMany(mappedBy = "commande")
    @Cascade(value = { CascadeType.ALL })
    @SortNatural
    @MapKey(name = "article")
    private SortedMap<Article, LigneCommande> detailsCommandes = new TreeMap<>();

    public Commande() {
        super();
    }

    public Commande(int numCommande, Client client, Date dateCommande, String etat) {
        super();
        this.numCommande = numCommande;
        this.client = client;
        this.dateCommande = dateCommande;
        this.etat = etat;
    }

    public int getNumCommande() {
        return numCommande;
    }

    public void setNumCommande(int numCommande) {
        this.numCommande = numCommande;
    }

    public Client getClient() {
        return client;
    }

    public void setClient(Client client) {
        this.client = client;
    }

    public Date getDateCommande() {
        return dateCommande;
    }

    public void setDateCommande(Date dateCommande) {
        this.dateCommande = dateCommande;
    }

    public String getEtat() {
        return etat;
    }

    public void setEtat(String etat) {
        this.etat = etat;
    }

}
```

Structure de Projet Front-end



Partie User Interface

➢ Page de présentation:



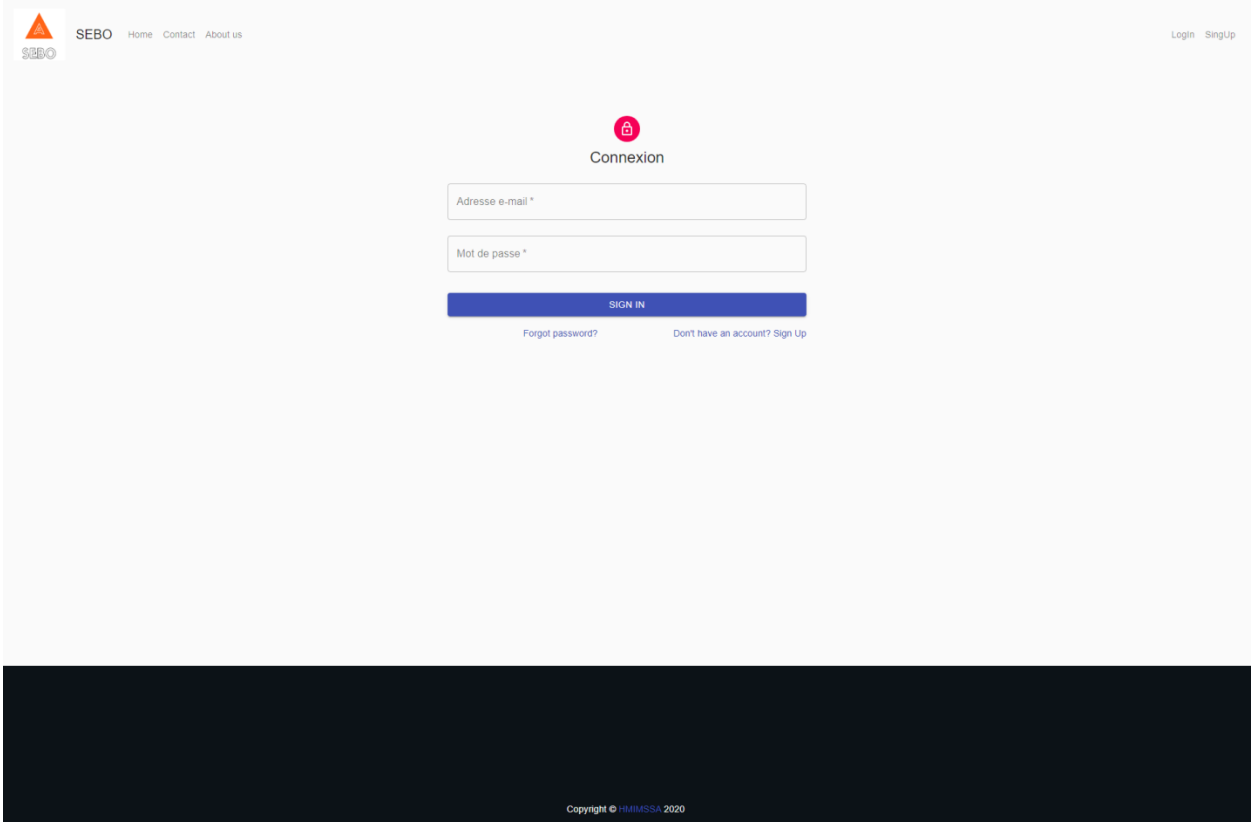
Code :

```
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: {
        paddingTop: theme.spacing(8),
        paddingBottom: theme.spacing(8),
    },
    card: {
        height: '100%',
        display: 'flex',
        flexDirection: 'column',
    },
    cardMedia: {
        paddingTop: '56.25%', // 16:9
    },
    cardContent: {
        flexGrow: 1,
    },

    button: {
        padding: "15px 25px",
        fontSize: "24px",
        cursor: "pointer",
        textAlign: "center",
        outline: "none",
        color: "#fff0",
        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>
```

```
                    <img
                        className="d-block w-100"
                        src={home_banner}
                        alt="First slide"
                    />
                    <Carousel.Caption>
                        <h3>First slide label</h3>
                        <p>Nulla vitae elit libero, a pharetra augue mollis interdum.</p>
                    </Carousel.Caption>
                </Carousel.Item>
                <Carousel.Item>
                    <img
                        className="d-block w-100"
                        src={home_banner2}
                        alt="First slide"
                    />
                    <Carousel.Caption>
                        <h3>First slide label</h3>
                        <p>Nulla vitae elit libero, a pharetra augue mollis interdum.</p>
                    </Carousel.Caption>
                </Carousel.Item>
                <Carousel.Item>
                    <img
                        className="d-block w-100"
                        src={home_banner3}
                        alt="First slide"
                    />
                    <Carousel.Caption>
                        <h3>First slide label</h3>
                        <p>Nulla vitae elit libero, a pharetra augue mollis interdum.</p>
                    </Carousel.Caption>
                </Carousel.Item>
            </Carousel>
            <div style={{
                background: "#0c2c39"

            }} className={classes.heroContent}>
                <Container maxWidth="md" >

                    <Typography style={{ color: "#fff" }} component="h1" variant="h2" align="center" gutterBottom
>
                        Bienvenue Chez SEBO
                    </Typography>
                    <Typography style={{ color: "#fff" }} variant="h5" align="center" paragraph>
                        Découvrez, achetez la musique de vos artistes préférés sur SEBO..
                        </Typography>
                    <div className={classes.heroButtons}>
                        <Grid container spacing={2} justify="center">
                            <Grid item>
                                <Link style={{ textDecoration: "none", color: "#fff" }}
                                    className={classes.button}
                                    variant="contained"
                                    color="primary"
                                    to="/SignUp">Inscription</Link>
                            </Grid>
                            <Grid item>
                                <Link style={{ textDecoration: "none", color: "#fff" }}
                                    className={classes.button} to="/SignIn">Connexion</Link>
                            </Grid>
                        </Grid>
                    </div>
                </Container>
            </div>

        </div >
    );
}
```

➢ Page d'authentification :

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

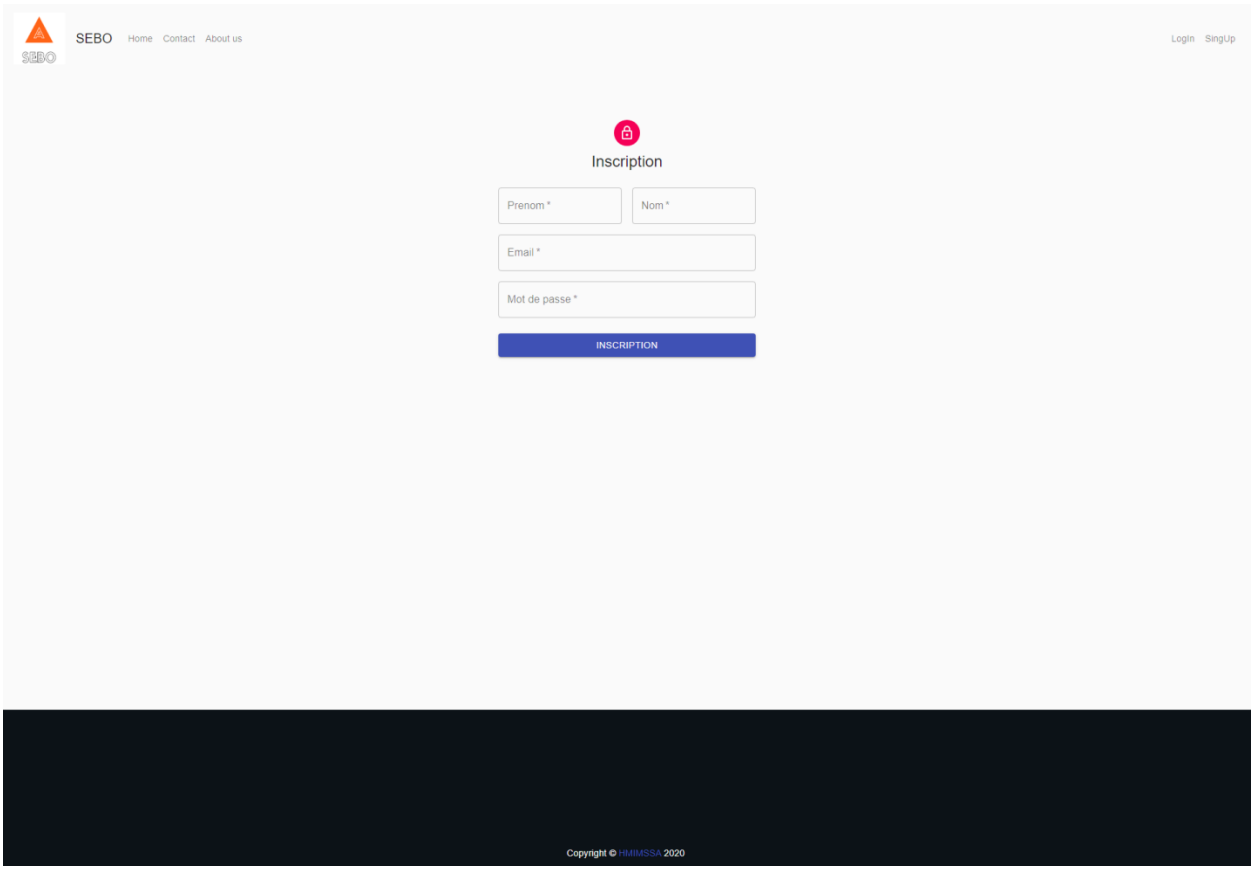



```
import React, { useState } from 'react';
import Avatar from '@material-ui/core/Avatar';
import Button from '@material-ui/core/Button';
import CssBaseline from '@material-ui/core/CssBaseline';
import TextField from '@material-ui/core/TextField';
import Link from '@material-ui/core/Link';
import Grid 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);
                            }}
```

```
                        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:



Code :

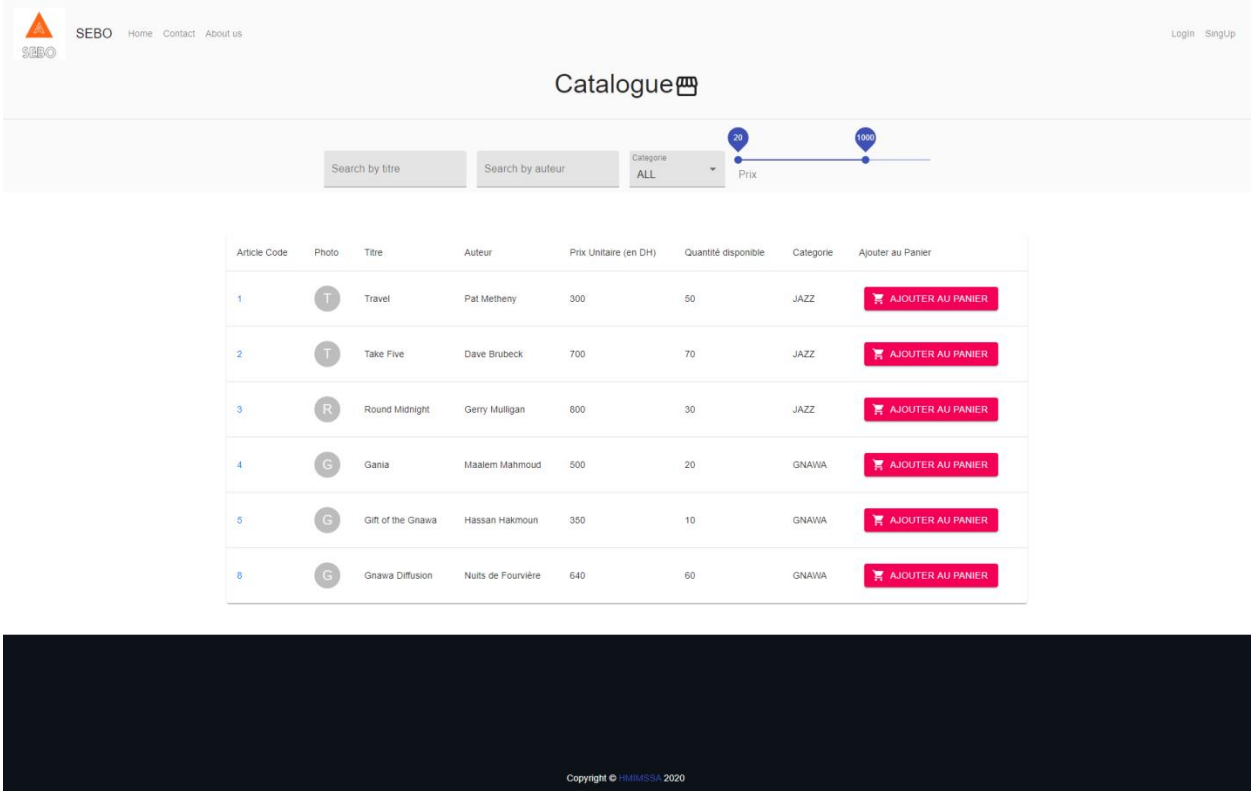
```
import React, { useState } from 'react';
import Avatar from '@material-ui/core/Avatar';
import Button from '@material-ui/core/Button';
import CssBaseline from '@material-ui/core/CssBaseline';
import TextField from '@material-ui/core/TextField';
import Grid from '@material-ui/core/Grid';
import LockOutlinedIcon 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 :

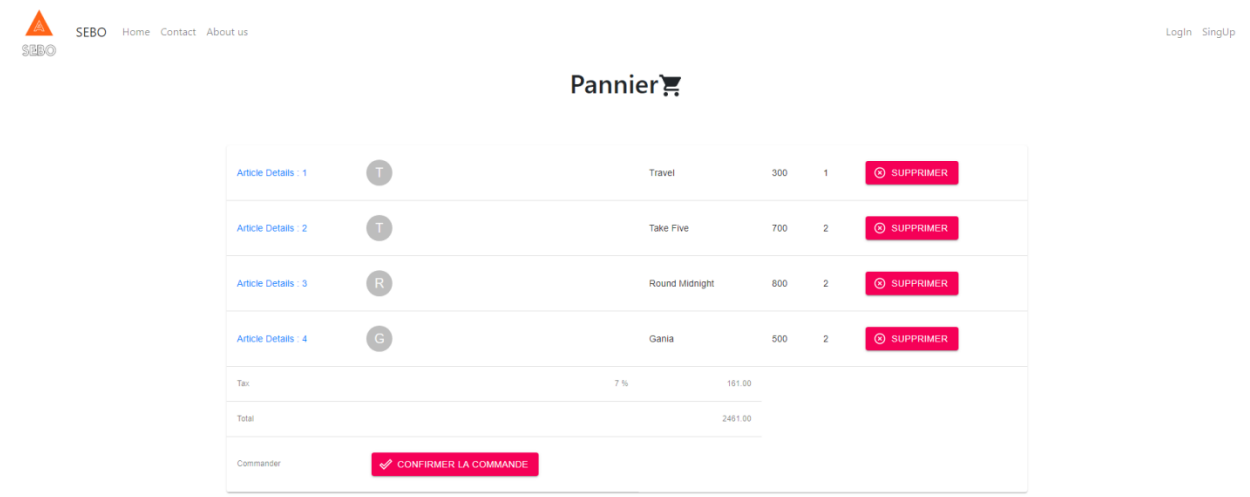
```
import React, { useState, useEffect, useRef } from 'react';
import Table from '@material-ui/core/Table';
import TableBody from '@material-ui/core/TableBody';
import TableCell from '@material-ui/core/TableCell';
import TableContainer from '@material-ui/core/TableContainer';
import TableHead from '@material-ui/core/TableHead';
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>
        <div className={classes.heroContent}>
            <Container maxWidth="lg"  >
                <TableContainer component={Paper} >
                    <Table className={classes.table} aria-label="simple table">
                        <TableHead>
                            <TableRow>
                                <TableCell>Article Code</TableCell>
                                <TableCell >Photo</TableCell>
                                <TableCell >Titre</TableCell>
                                <TableCell >Auteur</TableCell>
                                <TableCell >Prix Unitaire (en DH)</TableCell>
                                <TableCell >Quantité disponible</TableCell>
                                <TableCell >Categorie</TableCell>
                                <TableCell >Ajouter au Panier</TableCell>
                            </TableRow>
                        </TableHead>
                        <TableBody>
                            {articlesCopy?.filter(row => (
                                row.titre.toLowerCase().search(filterBytitre) !== -1) &&
                                (row.auteur.toLowerCase().search(filterByAuteur) !== -1)).map(row => (
                                    <TableRow key={row.codeArticle}>
                                        <TableCell component="th" scope="row">
                                            <Link
                                                to={`/ArticleDetails/${row.codeArticle}`}
                                            >  {row.codeArticle}</Link>
                                        </TableCell>
                                        <TableCell >
                                            <Avatar alt={row.titre} src="/static/images/avatar/1.jpg" />
                                        </TableCell>
                                        <TableCell >{row.titre}</TableCell>
                                        <TableCell >{row.auteur}</TableCell>
                                        <TableCell >{row.prix}</TableCell>
                                        <TableCell >{row.stock}</TableCell>
                                        <TableCell>{row.categorie?.cat}</TableCell>
                                        <TableCell >
                                            <Button
                                                variant="contained"
                                                color="secondary"
                                                name={row.codeArticle}
                                                className={classes.button}
                                                onClick={handleClick}
                                                startIcon={<ShoppingCartIcon />}
                                            >
                                                Ajouter au panier
                                                </Button>
                                        </TableCell>
                                    </TableRow>
                                ))}
                        </TableBody>
                    </Table>
                </TableContainer>
            </Container>
        </div>


    </main>

  );

}
export default withStyles(styles)(Catalogue);
```

➢ Page Pannier:



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,
```

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

const TAX_RATE = 0.07;


function ccyFormat(num) {
    return `${num.toFixed(2)}`;
}

function priceRow(qty, unit) {
    return qty * unit;
}

function createRow(desc, qty, unit) {
    const price = priceRow(qty, unit);
    return { desc, qty, unit, price };
}

function subtotal(items) {
    //return items.map(({ price }) => price).reduce((sum, i) => sum + i, 0);
    return items?.map(item => item.article.prix).reduce((sum, i) => sum + i, 0);
}



function Pannier(props) {
    const { classes } = props;
    const oClient = props.location.oClient;

    const invoiceSubtotal = subtotal(oClient?.commandes[0]?.lignesCommande);
    const invoiceTaxes = TAX_RATE * invoiceSubtotal;
    const invoiceTotal = invoiceTaxes + invoiceSubtotal;

    const handleClick = (event) => {


    };
    useEffect(() => {
        console.log(oClient);
    }, [oClient]);


    return (
        <main style={{ minHeight: "100%" }}>
            <div>
                <h1>
                    <span style={{ verticalAlign: "bottom" }}>Pannier</span>
                    <ShoppingCartIcon fontSize="inherit" />
                </h1>
            </div>
            <div className={classes.heroContent}>
                <Container maxWidth="lg"  >
                    <TableContainer component={Paper} >
                        <Table className={classes.table} aria-label="simple table">

                            <TableBody>
                                {oClient?.commandes?.map(row => (
                                    row.lignesCommande.map(linge => (
```

```
                                    <TableRow key={linge.lignesCommande}>
                                        <TableCell component="th" scope="row">
                                            <Link
                                                to={`/ArticleDetails/${linge.article.codeArticle}`}
                                            > Article Details : {linge.article.codeArticle}</Link>
                                        </TableCell>
                                        <TableCell >
                                            <Avatar alt={linge.article.titre}
                                                src="/static/images/avatar/1.jpg" />
                                        </TableCell>
                                        <TableCell >{linge.article.titre}</TableCell>
                                        <TableCell >{linge.article.prix}</TableCell>
                                        <TableCell >{linge.qteCde}</TableCell>
                                        <TableCell >
                                            <Button
                                                variant="contained"
                                                color="secondary"
                                                name={linge.article.codeArticle}
                                                className={classes.button}
                                                onClick={handleClick}
                                                startIcon={<HighlightOffIcon />}
                                            >
                                                Supprimer
                                                </Button>
                                        </TableCell>
                                    </TableRow>
                                ))

                            ))}
                        </TableBody>
                        <TableFooter  >

                            <TableRow>
                                <TableCell>Tax</TableCell>
                                <TableCell align="right">{`${(TAX_RATE * 100).toFixed(0)} %`}</TableCell>
                                <TableCell align="right">{ccyFormat(invoiceTaxes)}</TableCell>
                            </TableRow>
                            <TableRow>
                                <TableCell colSpan={2}>Total</TableCell>
                                <TableCell align="right">{ccyFormat(invoiceTotal)}</TableCell>
                            </TableRow>
                            <TableRow variant="footer">
                                <TableCell  >Commander</TableCell>
                                <TableCell variant="footer">
                                    <Button
                                        variant="contained"
                                        color="secondary"

                                        className={classes.button}
                                        onClick={handleClick}
                                        startIcon={<DoneOutlineIcon />}
                                    >
                                        confirmer  la commande
                                                </Button>
                                </TableCell>
                            </TableRow></TableFooter>

                    </Table>
                </TableContainer>
            </Container>
        </div>



    </main>

  );

}
export default withStyles(styles)(Pannier);
```