MARRAKECH

جامعة القاضي عياض
UNIVERSITÉ CADI AYYAD

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

Micro Projet

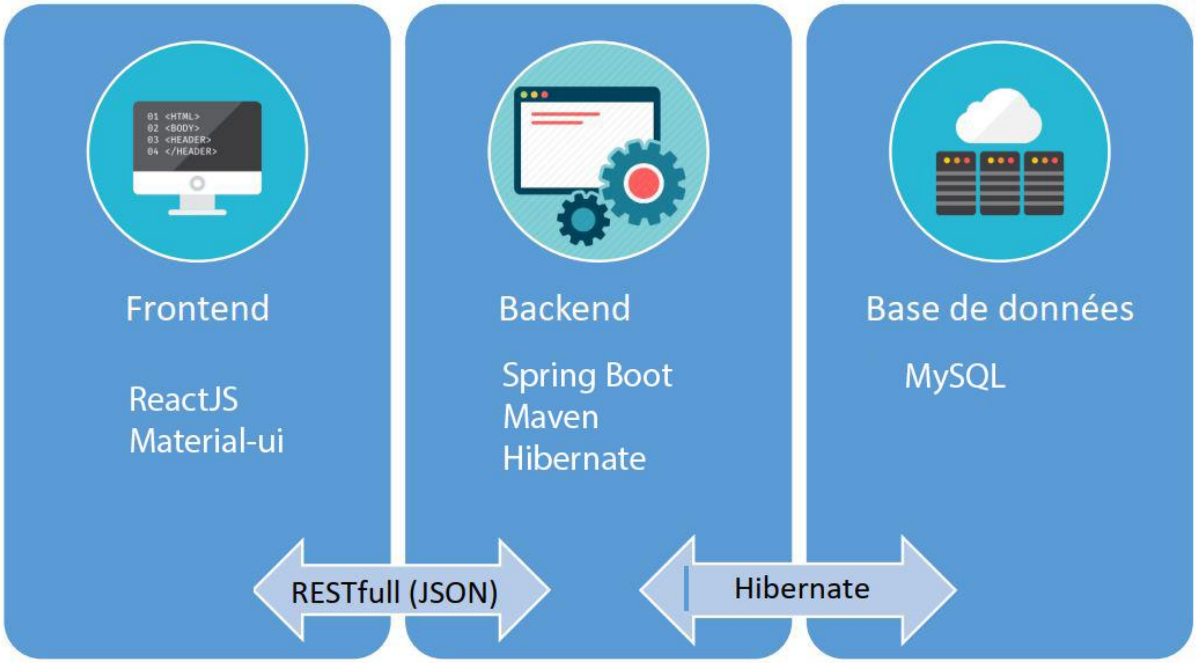
Réalisation d'un site web pour l'achat de la musique .avec Spring Boot , Maven ,ReactJs ,Material-UI et MySQL ..

Hmimssa Soufiane

2020/2021

Objectif :

Réalisation d'un site web pour l’achat de la musique . l'architecture utilise tout au long du developpement est divise en trois parties.

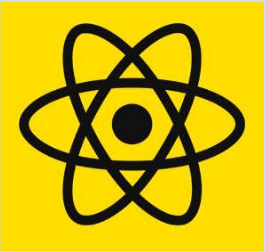


Les outils de développement utilisés :

- Material UI :

- Spring Boot :

- ReactJS :



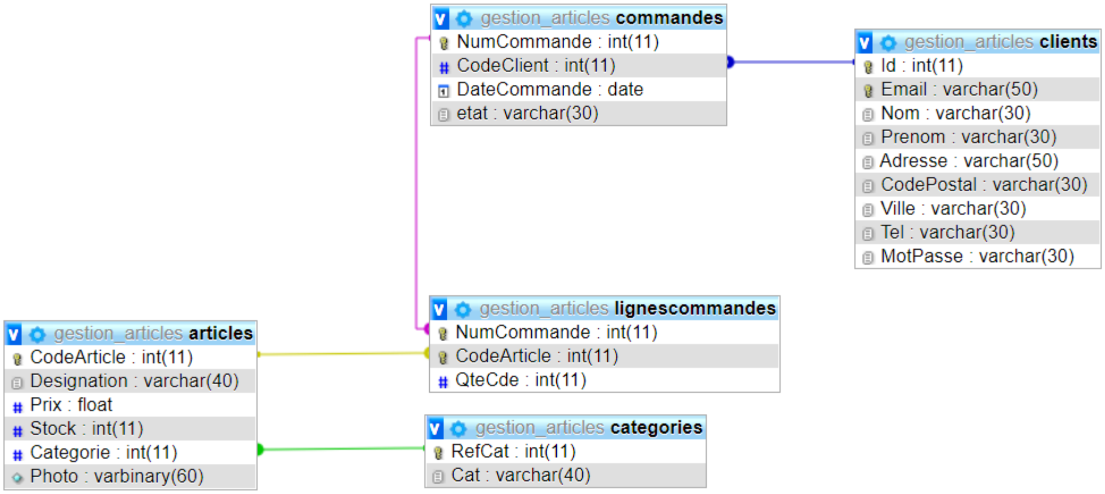
- MySql :

- Hibernate :

- Maven:

Notre base de données

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



gestion_articles **commandes**
NumCommande : int(11)
CodeClient : int(11)
DateCommande : date
etat : varchar(30)

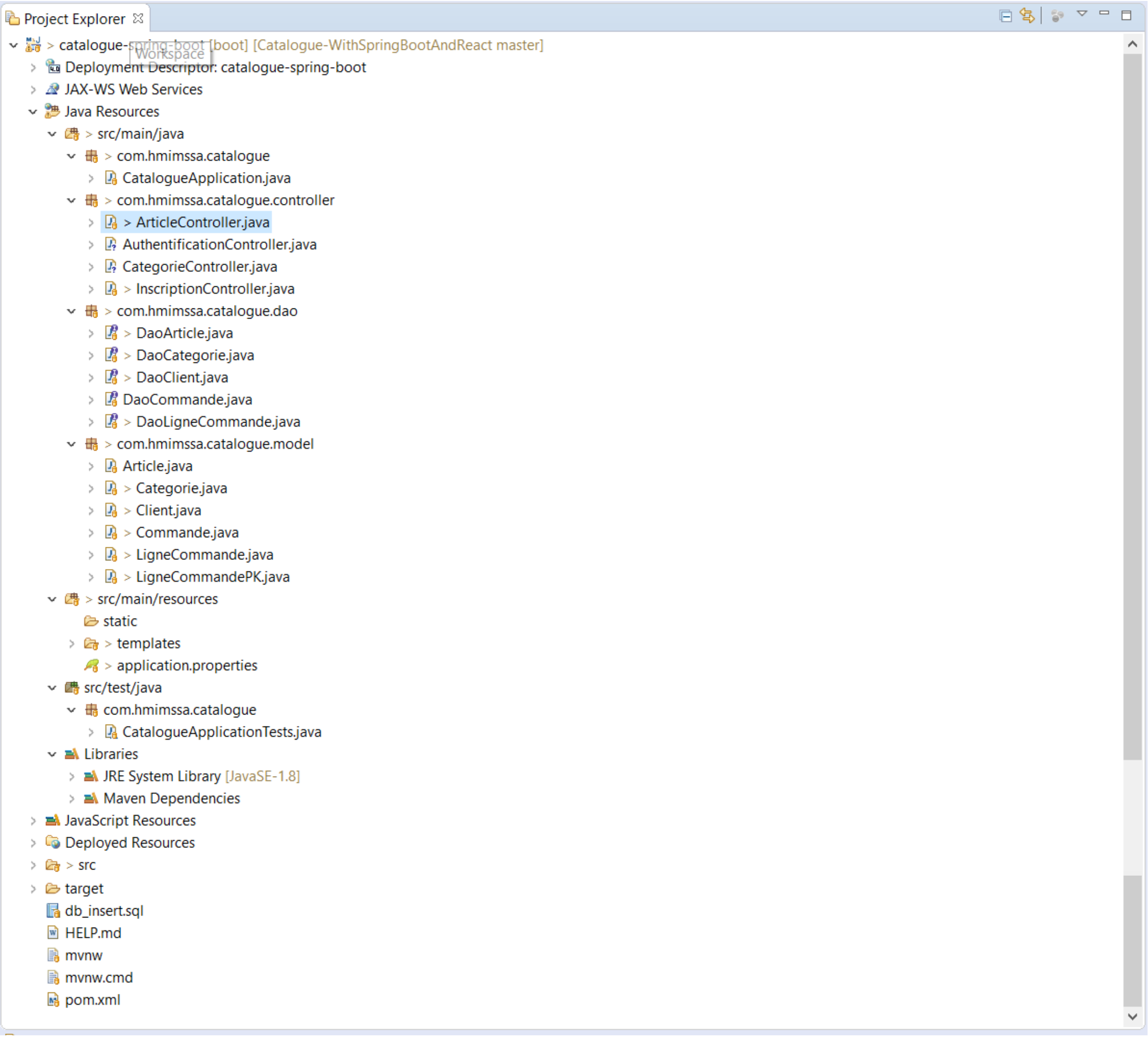
gestion_articles **clients**
Id : int(11)
Email : varchar(50)
Nom : varchar(30)
Prenom : varchar(30)
Adresse : varchar(50)
CodePostal : varchar(30)
Ville : varchar(30)
Tel : varchar(30)
MotPasse : varchar(30)

gestion_articles **lignescommandes**
NumCommande : int(11)
CodeArticle : int(11)
QteCde : int(11)

gestion_articles **articles**
CodeArticle : int(11)
Designation : varchar(40)
Prix : float
Stock : int(11)
Categorie : int(11)
Photo : varbinary(60)

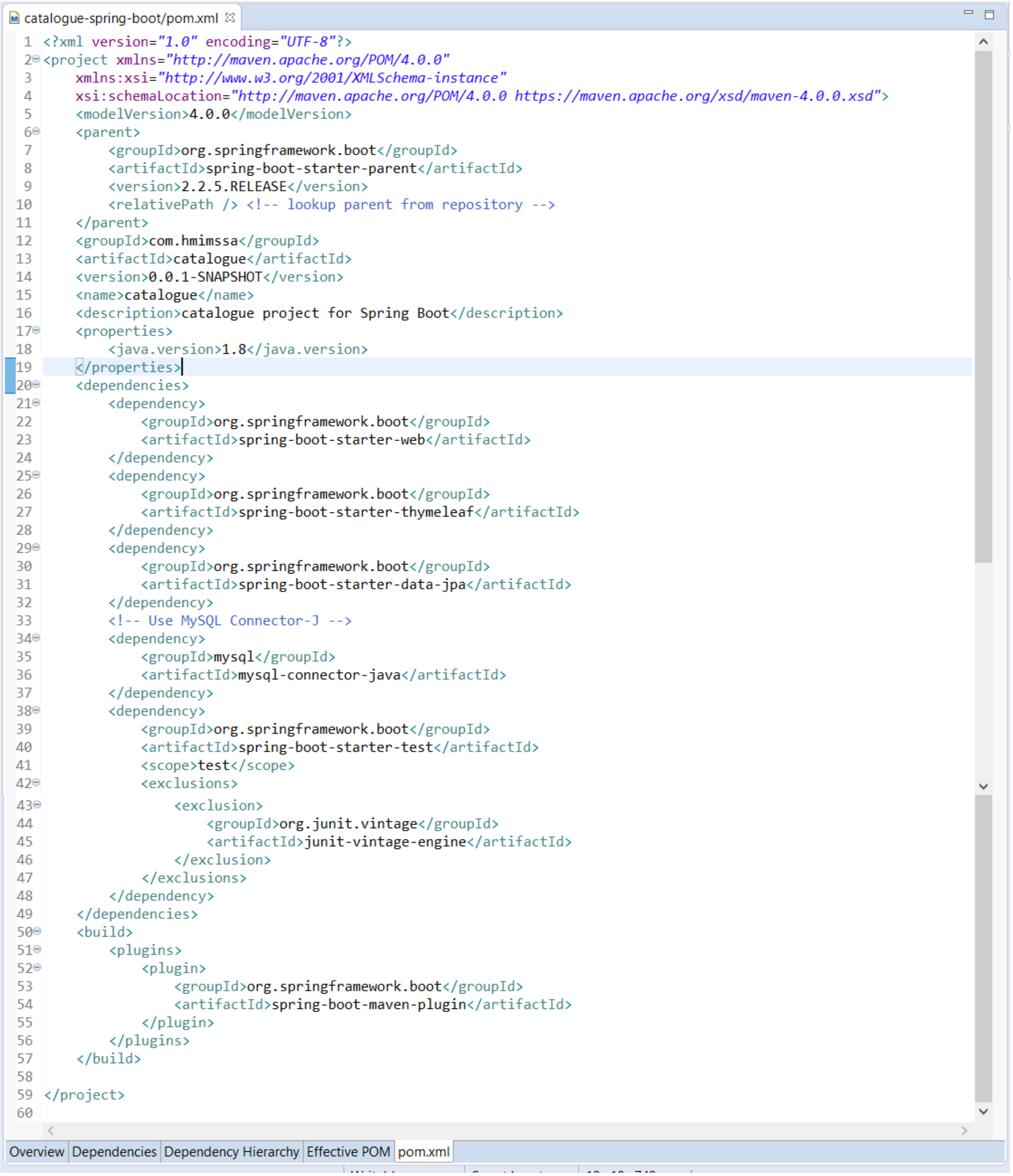
gestion_articles **categories**
RefCat : int(11)
Cat : varchar(40)

Structure de Projet Back-end



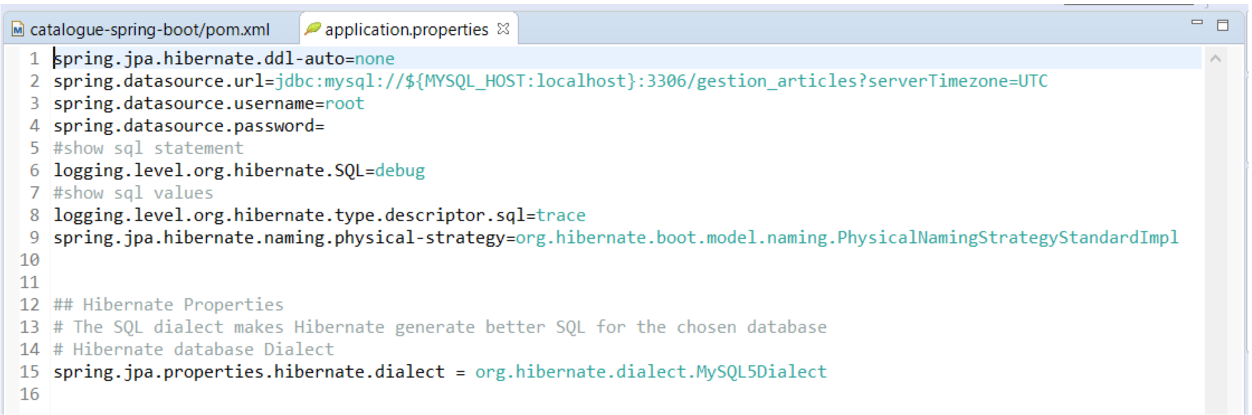
```
v > catalogue-spring-boot [boot] [Catalogue-WithSpringBootAndReact master]
  > Deployment Descriptor: catalogue-spring-boot
  > JAX-WS Web Services
  v Java Resources
    v > src/main/java
      v > com.hmimssa.catalogue
        > CatalogueApplication.java
      v > com.hmimssa.catalogue.controller
        > > ArticleController.java
        > AuthentificationController.java
        > CategorieController.java
        > > InscriptionController.java
      v > com.hmimssa.catalogue.dao
        > > DaoArticle.java
        > > DaoCategorie.java
        > > DaoClient.java
        > DaoCommande.java
        > > DaoLigneCommande.java
      v > com.hmimssa.catalogue.model
        > Article.java
        > > Categorie.java
        > > Client.java
        > > Commande.java
        > > LigneCommande.java
        > > LigneCommandePK.java
    v > src/main/resources
        static
      > > templates
        > application.properties
    v src/test/java
      v com.hmimssa.catalogue
        > CatalogueApplicationTests.java
    v Libraries
      > JRE System Library [JavaSE-1.8]
      > Maven Dependencies
  > JavaScript Resources
  > Deployed Resources
  > > src
  > target
    db_insert.sql
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
```

Configuration du fichier POM.XML



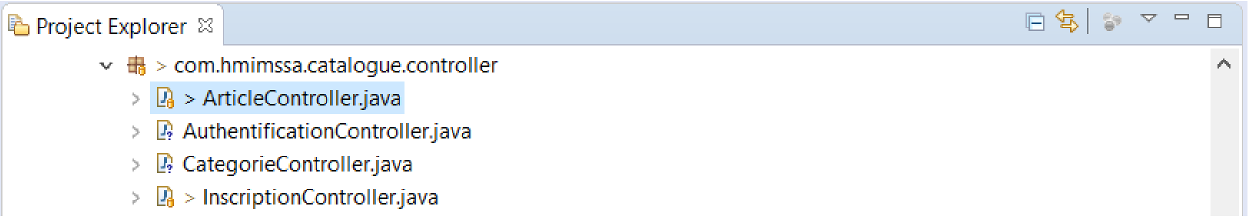
```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <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>

```

Configuration du fichier application.properties



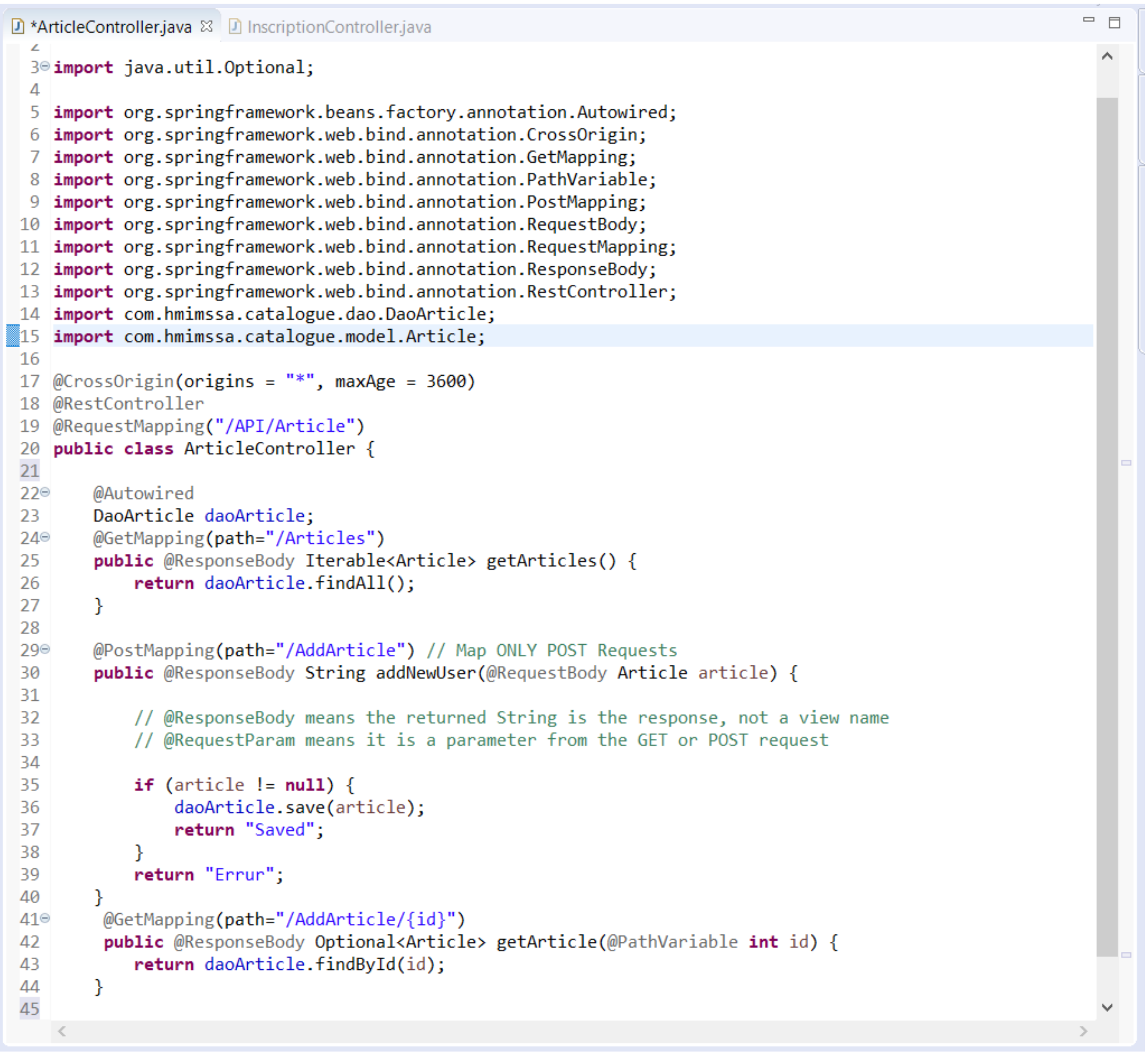
```
spring.jpa.hibernate.ddl-auto=none
spring.datasource.url=jdbc:mysql://${MYSQL_HOST:localhost}:3306/gestion_articles?serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=
#show sql statement
logging.level.org.hibernate.SQL=debug
#show sql values
logging.level.org.hibernate.type.descriptor.sql=trace
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl


## Hibernate Properties
# The SQL dialect makes Hibernate generate better SQL for the chosen database
# Hibernate database Dialect
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect

```

Couche Controller



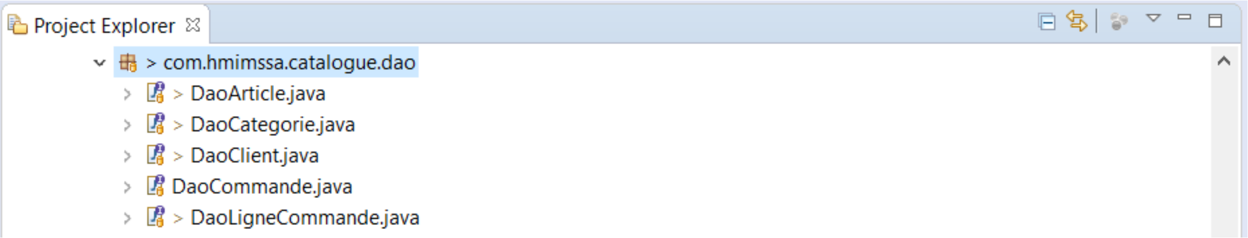
```
> com.hmimssa.catalogue.controller
    > ArticleController.java
    AuthentificationController.java
    CategorieController.java
    > InscriptionController.java
```

Exemple contrôleur d'Article :



```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);
    }

```

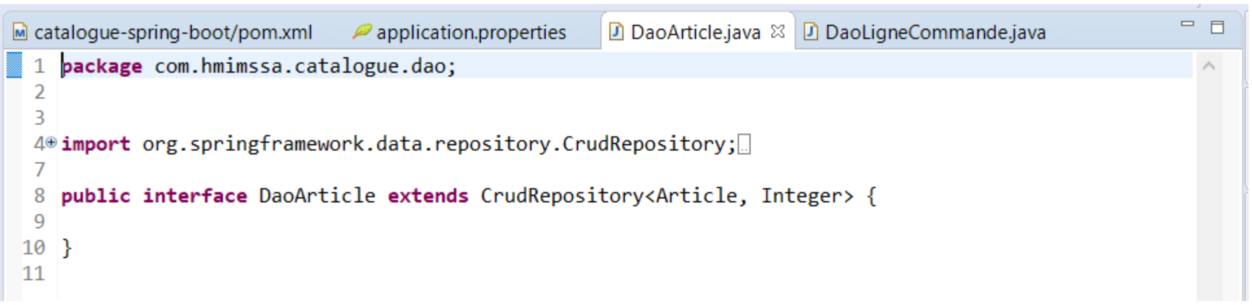
Couche Dao



Project Explorer

```
> com.hmimssa.catalogue.dao
    > DaoArticle.java
    > DaoCategorie.java
    > DaoClient.java
    DaoCommande.java
    > DaoLigneCommande.java
```

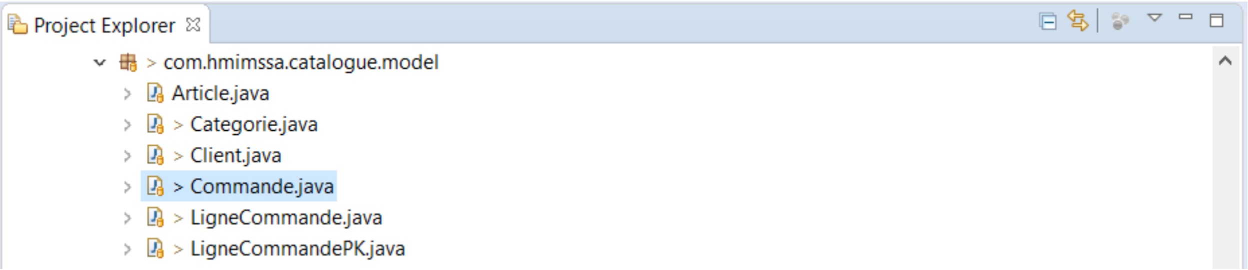
Exemple :



catalogue-spring-boot/pom.xml | application.properties | DaoArticle.java | DaoLigneCommande.java

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


import org.springframework.data.repository.CrudRepository;

public interface DaoArticle extends CrudRepository<Article, Integer> {

}
```

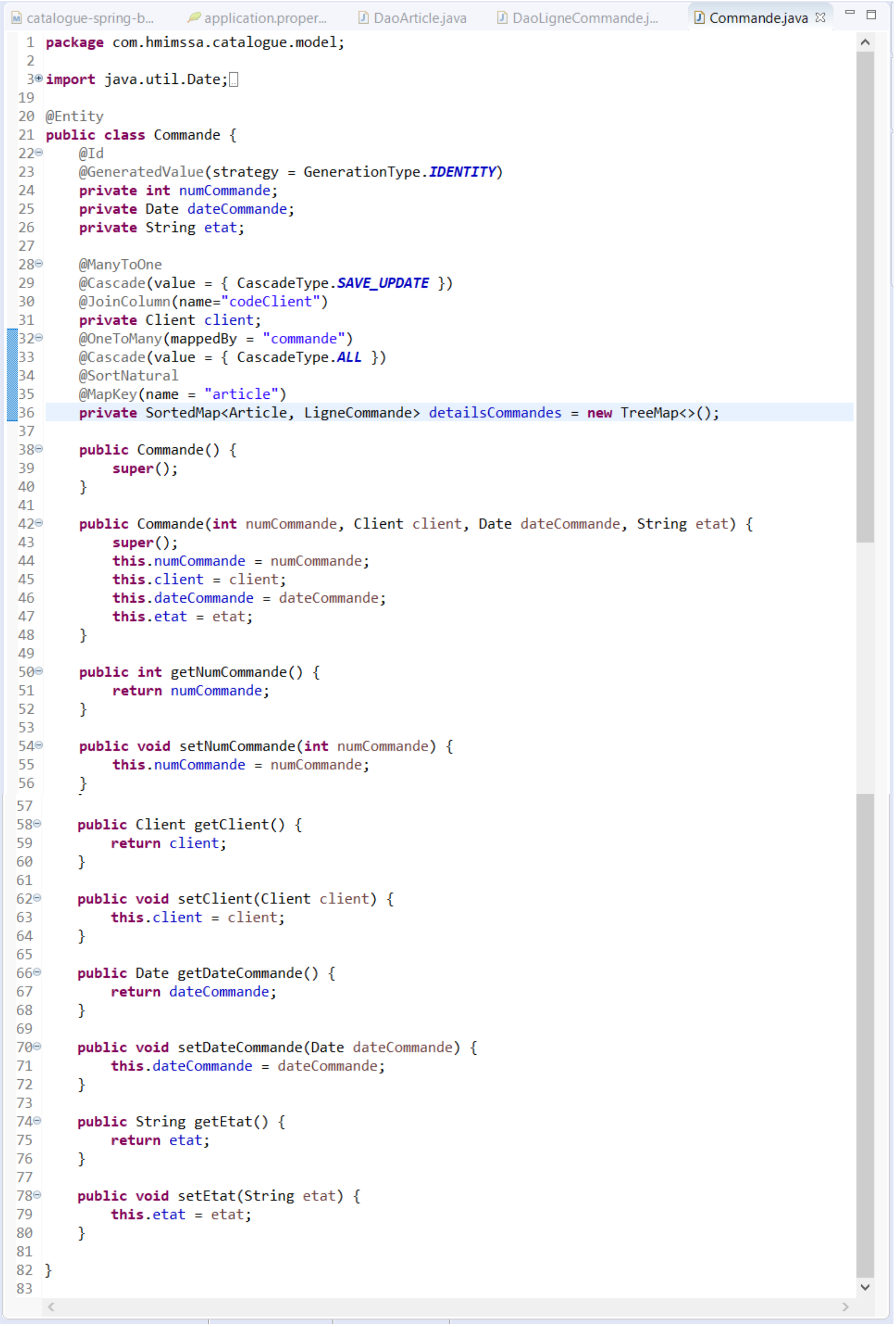
Couche Métier



Project Explorer

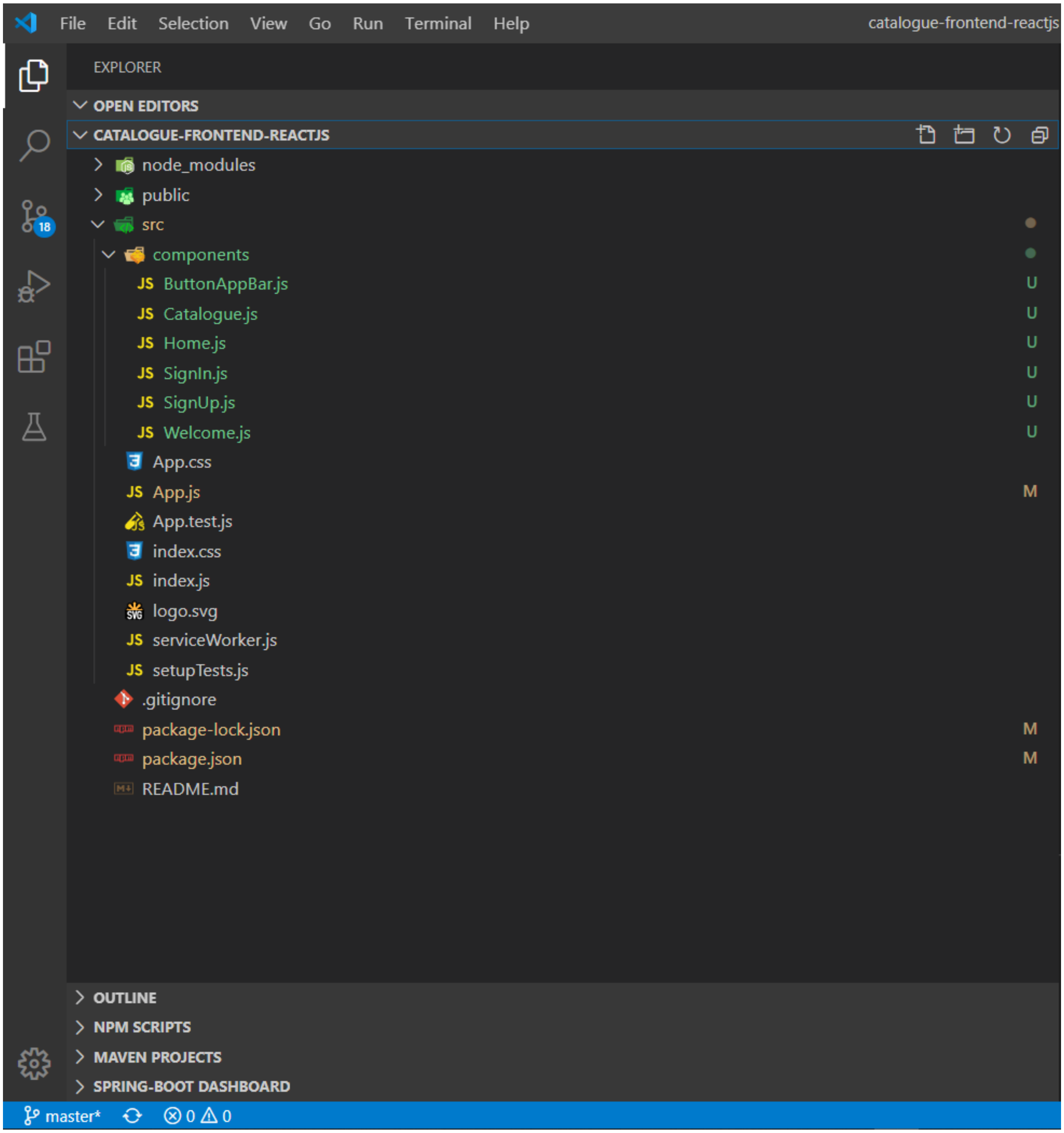
```
> com.hmimssa.catalogue.model
    Article.java
    > Categorie.java
    > Client.java
    > Commande.java
    > LigneCommande.java
    > LigneCommandePK.java
```

Exemple de class commande de couche metier



```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;
    }

}
```

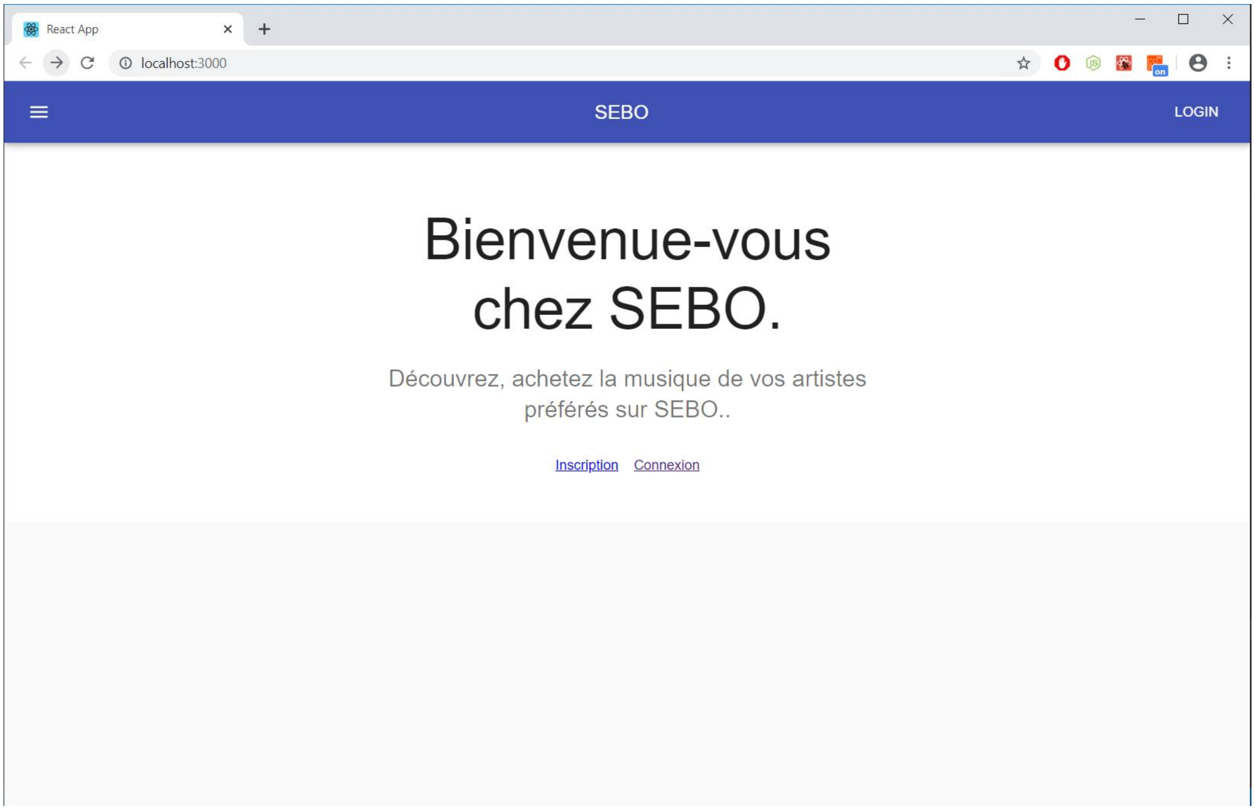
Structure de Projet Front-end



```
CATALOGUE-FRONTEND-REACTJS
  > node_modules
  > public
  v src
    v components
        ButtonAppBar.js      U
        Catalogue.js         U
        Home.js              U
        SignIn.js            U
        SignUp.js            U
        Welcome.js           U
      App.css
      App.js                 M
      App.test.js
      index.css
      index.js
      logo.svg
      serviceWorker.js
      setupTests.js
    .gitignore
    package-lock.json        M
    package.json             M
    README.md
```

Partie User Interface :

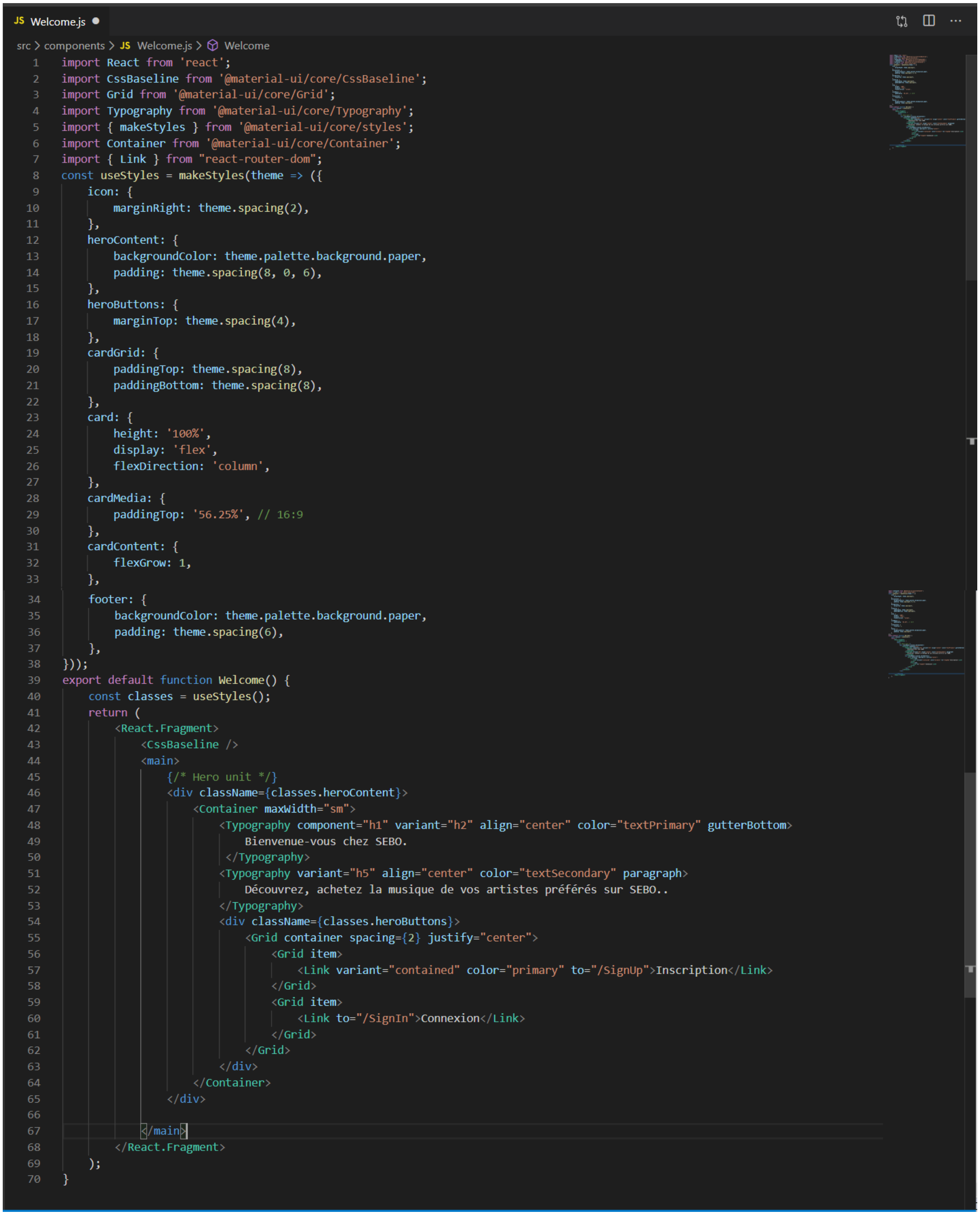
une page simple de présentation



Bienvenue-vous chez SEBO.

Découvrez, achetez la musique de vos artistes préférés sur SEBO..

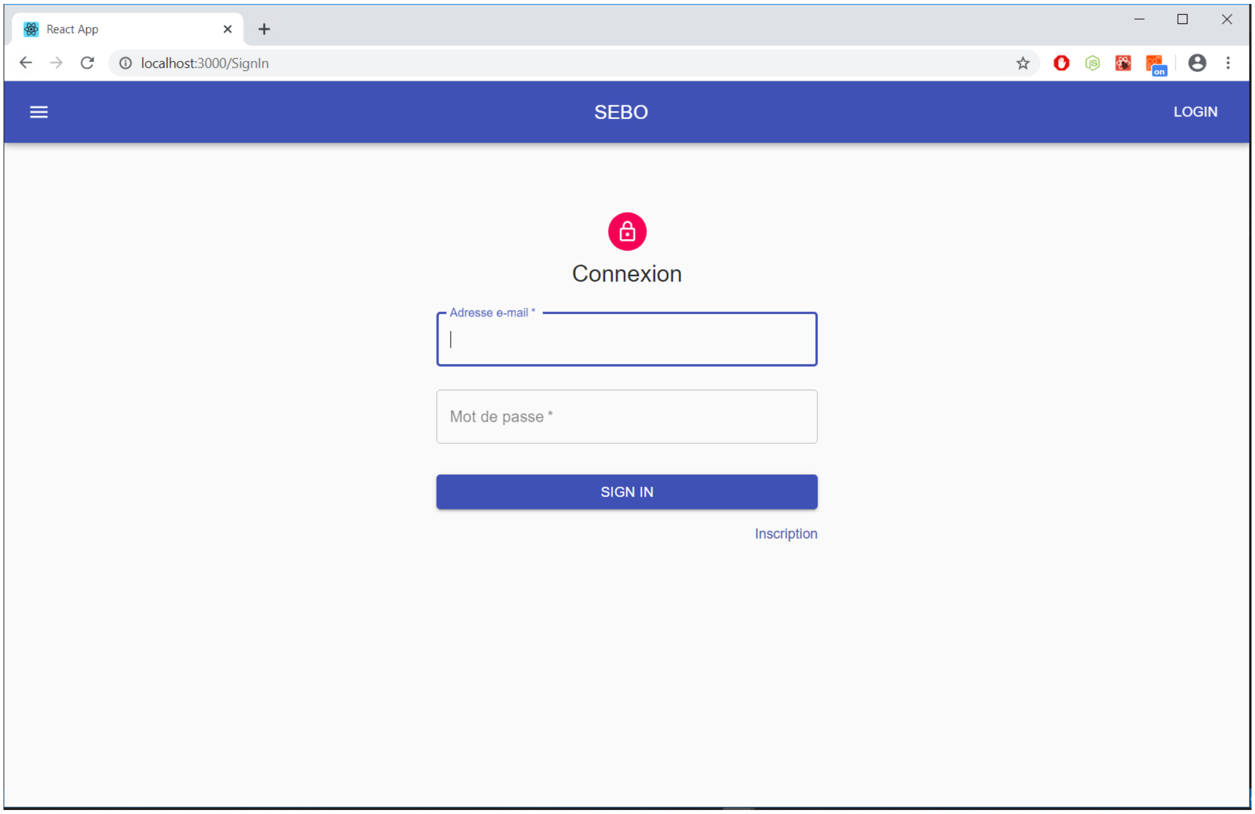
Code :

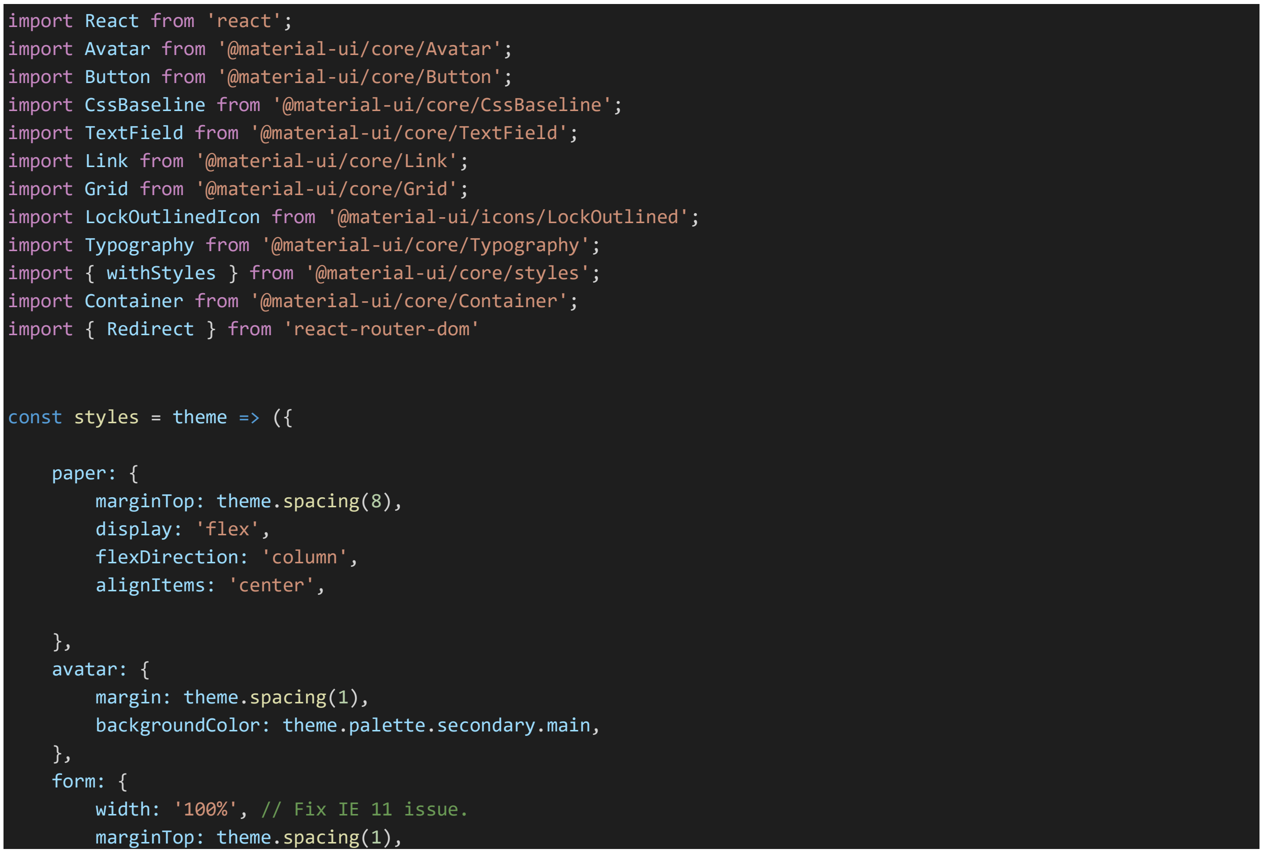


```
import React from 'react';
import CssBaseline from '@material-ui/core/CssBaseline';
import Grid from '@material-ui/core/Grid';
import Typography from '@material-ui/core/Typography';
import { makeStyles } from '@material-ui/core/styles';
import Container from '@material-ui/core/Container';
import { Link } from "react-router-dom";
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,
    },
    footer: {
        backgroundColor: theme.palette.background.paper,
        padding: theme.spacing(6),
    },
}));
export default function Welcome() {
    const classes = useStyles();
    return (
        <React.Fragment>
            <CssBaseline />
            <main>
                {/* Hero unit */}
                <div className={classes.heroContent}>
                    <Container maxWidth="sm">
                        <Typography component="h1" variant="h2" align="center" color="textPrimary" gutterBottom>
                            Bienvenue-vous chez SEBO.
                         </Typography>
                        <Typography variant="h5" align="center" color="textSecondary" 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 variant="contained" color="primary" to="/SignUp">Inscription</Link>
                                </Grid>
                                <Grid item>
                                    <Link to="/SignIn">Connexion</Link>
                                </Grid>
                            </Grid>
                        </div>
                    </Container>
                </div>

            </main>
        </React.Fragment>
    );
}
```

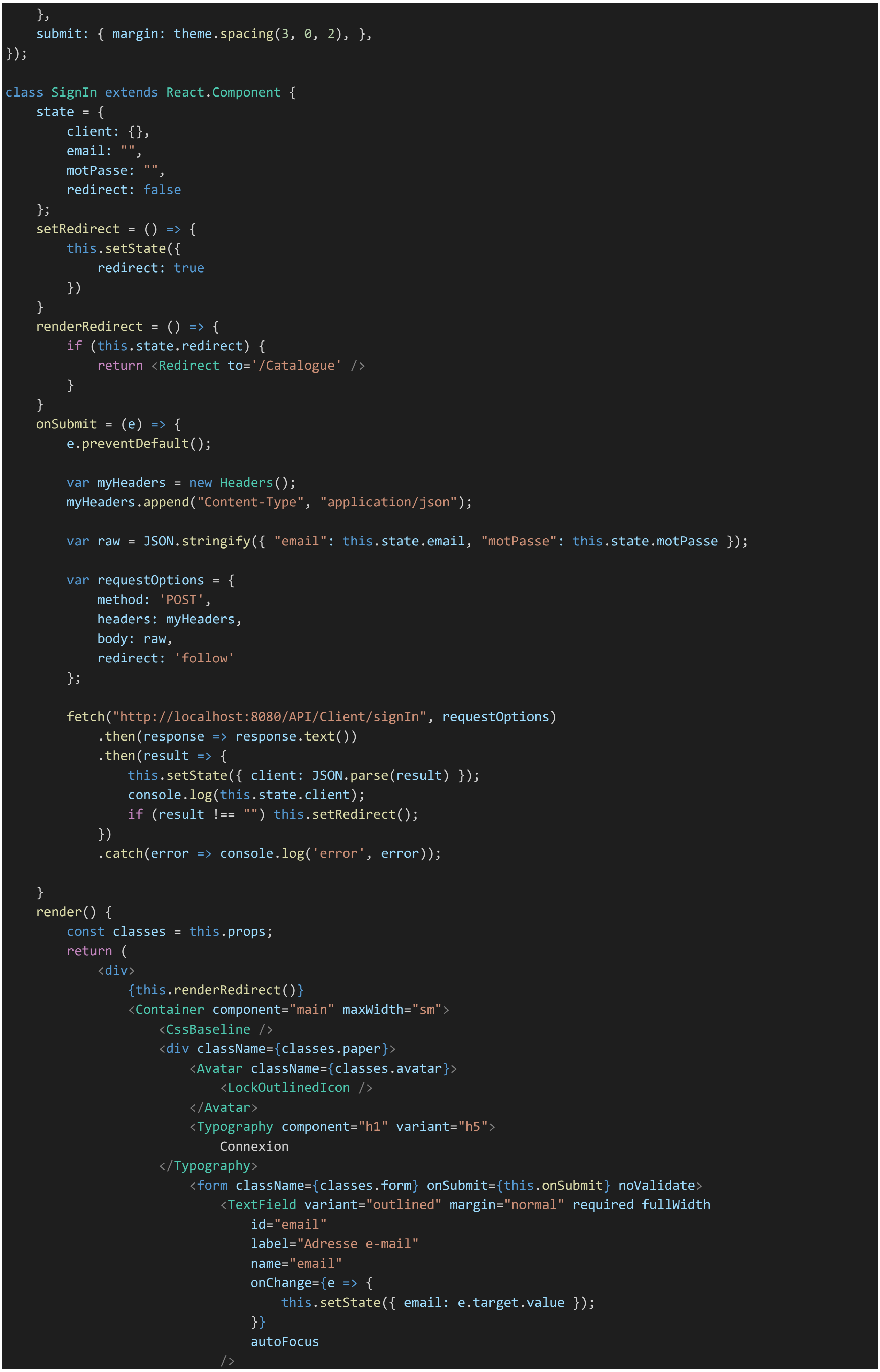
➢ Page d’authentification :

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

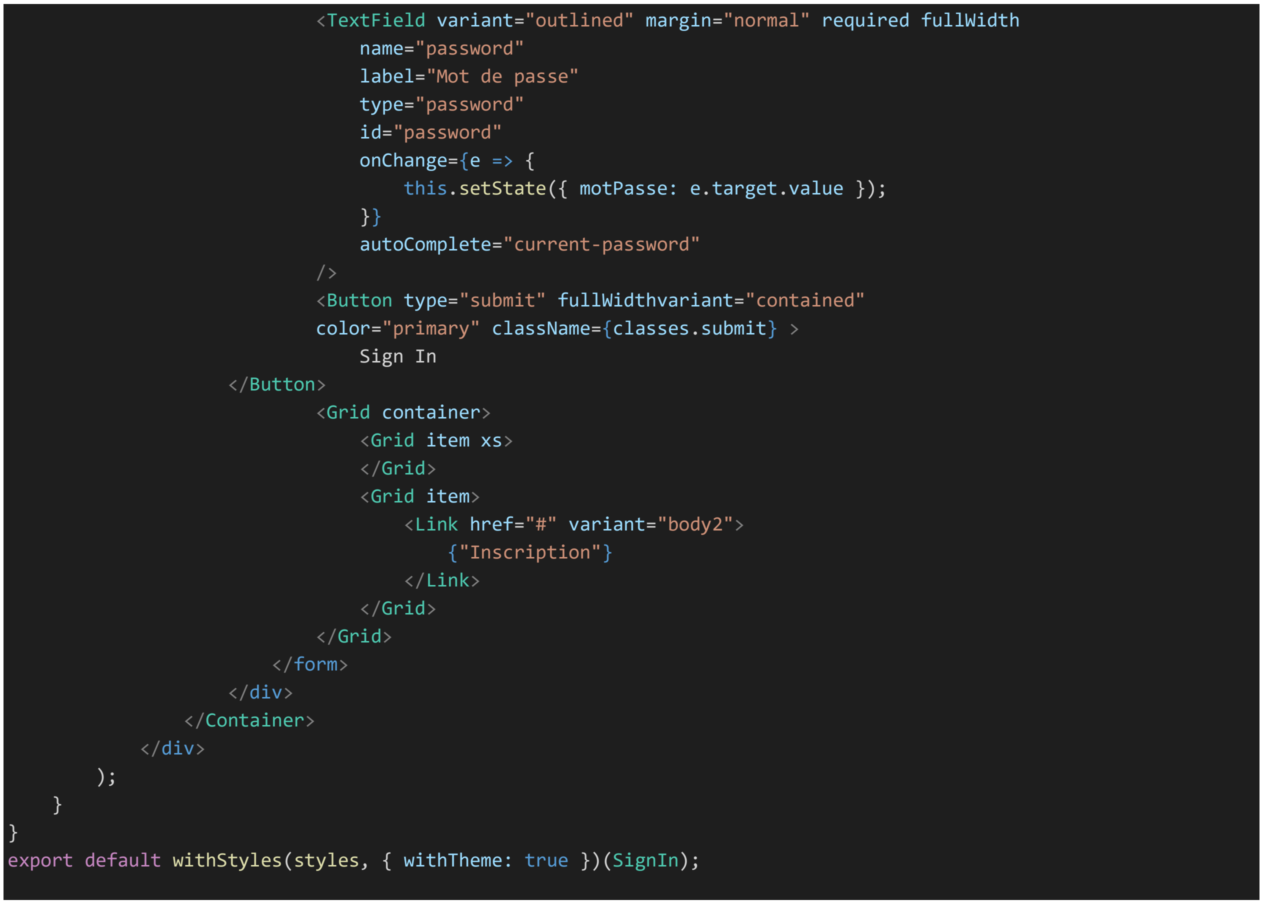




```
import React from 'react';
import Avatar from '@material-ui/core/Avatar';
import Button from '@material-ui/core/Button';
import CssBaseline from '@material-ui/core/CssBaseline';
import TextField from '@material-ui/core/TextField';
import Link from '@material-ui/core/Link';
import Grid from '@material-ui/core/Grid';
import LockOutlinedIcon from '@material-ui/icons/LockOutlined';
import Typography from '@material-ui/core/Typography';
import { withStyles } from '@material-ui/core/styles';
import Container from '@material-ui/core/Container';
import { Redirect } from 'react-router-dom'


const styles = 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),
```

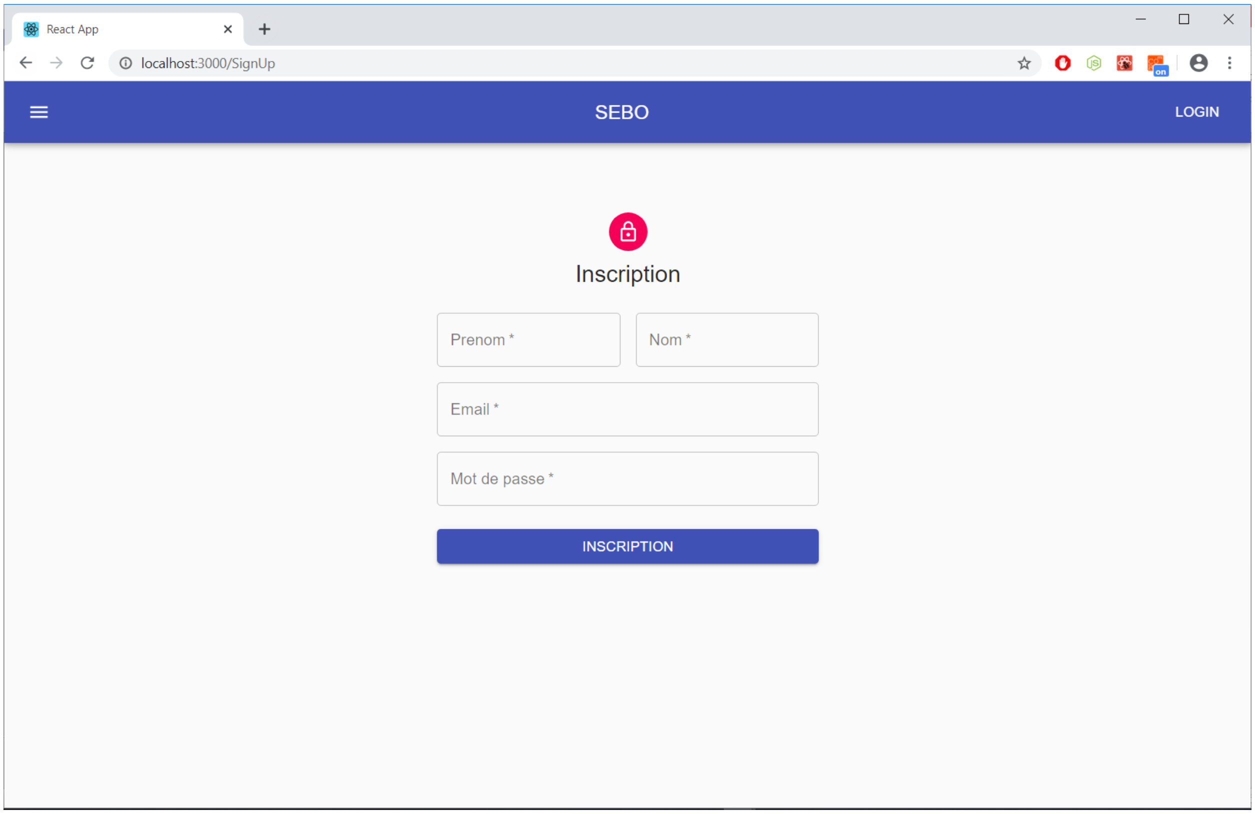


```
    },
    submit: { margin: theme.spacing(3, 0, 2), },
});

class SignIn extends React.Component {
    state = {
        client: {},
        email: "",
        motPasse: "",
        redirect: false
    };
    setRedirect = () => {
        this.setState({
            redirect: true
        })
    }
    renderRedirect = () => {
        if (this.state.redirect) {
            return <Redirect to='/Catalogue' />
        }
    }
    onSubmit = (e) => {
        e.preventDefault();

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

        var raw = JSON.stringify({ "email": this.state.email, "motPasse": this.state.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 => {
                this.setState({ client: JSON.parse(result) });
                console.log(this.state.client);
                if (result !== "") this.setRedirect();
            })
            .catch(error => console.log('error', error));

    }
    render() {
        const classes = this.props;
        return (
            <div>
                {this.renderRedirect()}
                <Container 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={this.onSubmit} noValidate>
                            <TextField variant="outlined" margin="normal" required fullWidth
                                id="email"
                                label="Adresse e-mail"
                                name="email"
                                onChange={e => {
                                    this.setState({ email: e.target.value });
                                }}
                                autoFocus
                            />
```

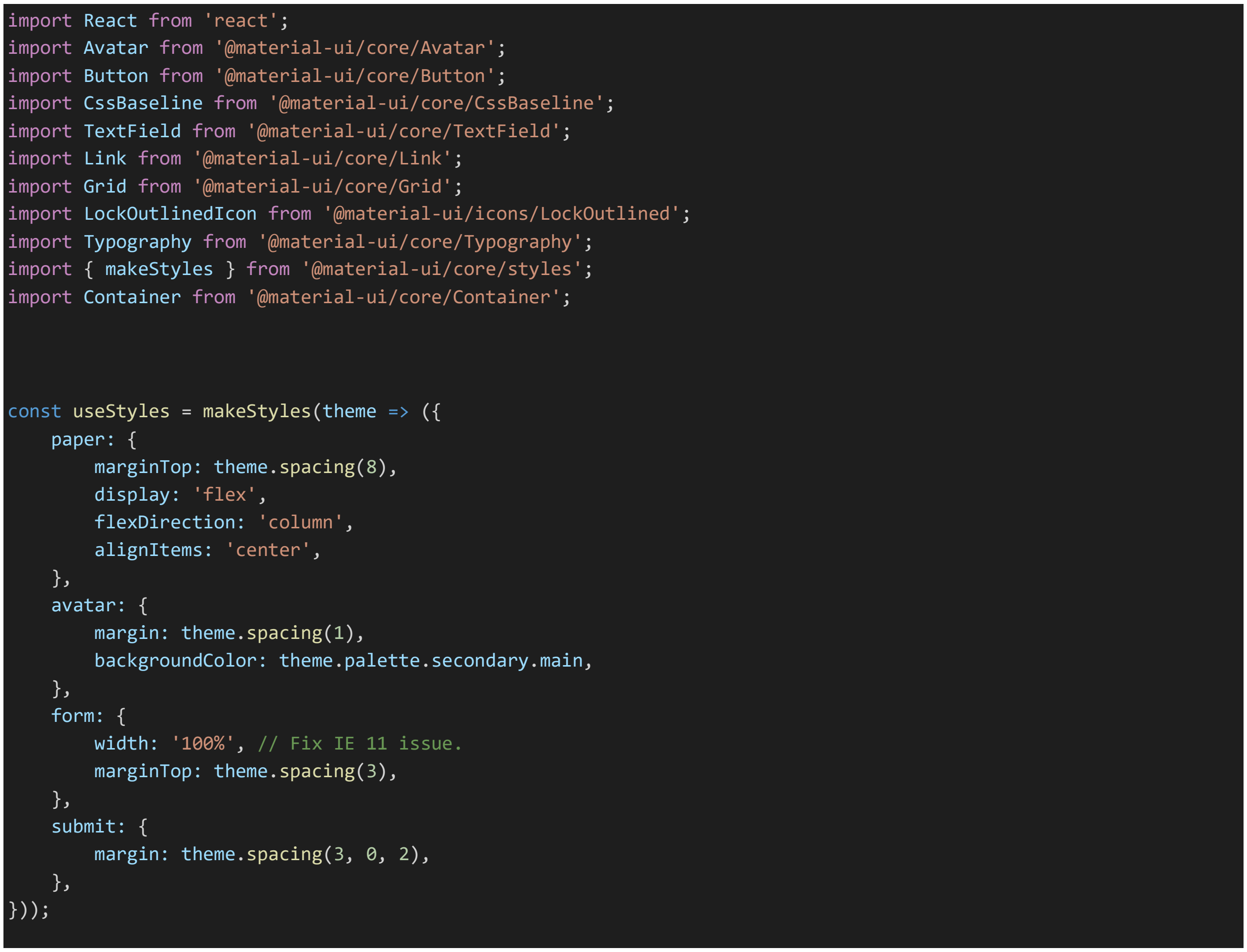


```
                            <TextField variant="outlined" margin="normal" required fullWidth
                                name="password"
                                label="Mot de passe"
                                type="password"
                                id="password"
                                onChange={e => {
                                    this.setState({ motPasse: e.target.value });
                                }}
                                autoComplete="current-password"
                            />
                            <Button type="submit" fullWidthvariant="contained"
                            color="primary" className={classes.submit} >
                                Sign In
                    </Button>
                            <Grid container>
                                <Grid item xs>
                                </Grid>
                                <Grid item>
                                    <Link href="#" variant="body2">
                                        {"Inscription"}
                                    </Link>
                                </Grid>
                            </Grid>
                        </form>
                    </div>
                </Container>
            </div>
        );
    }
}
export default withStyles(styles, { withTheme: true })(SignIn);
```

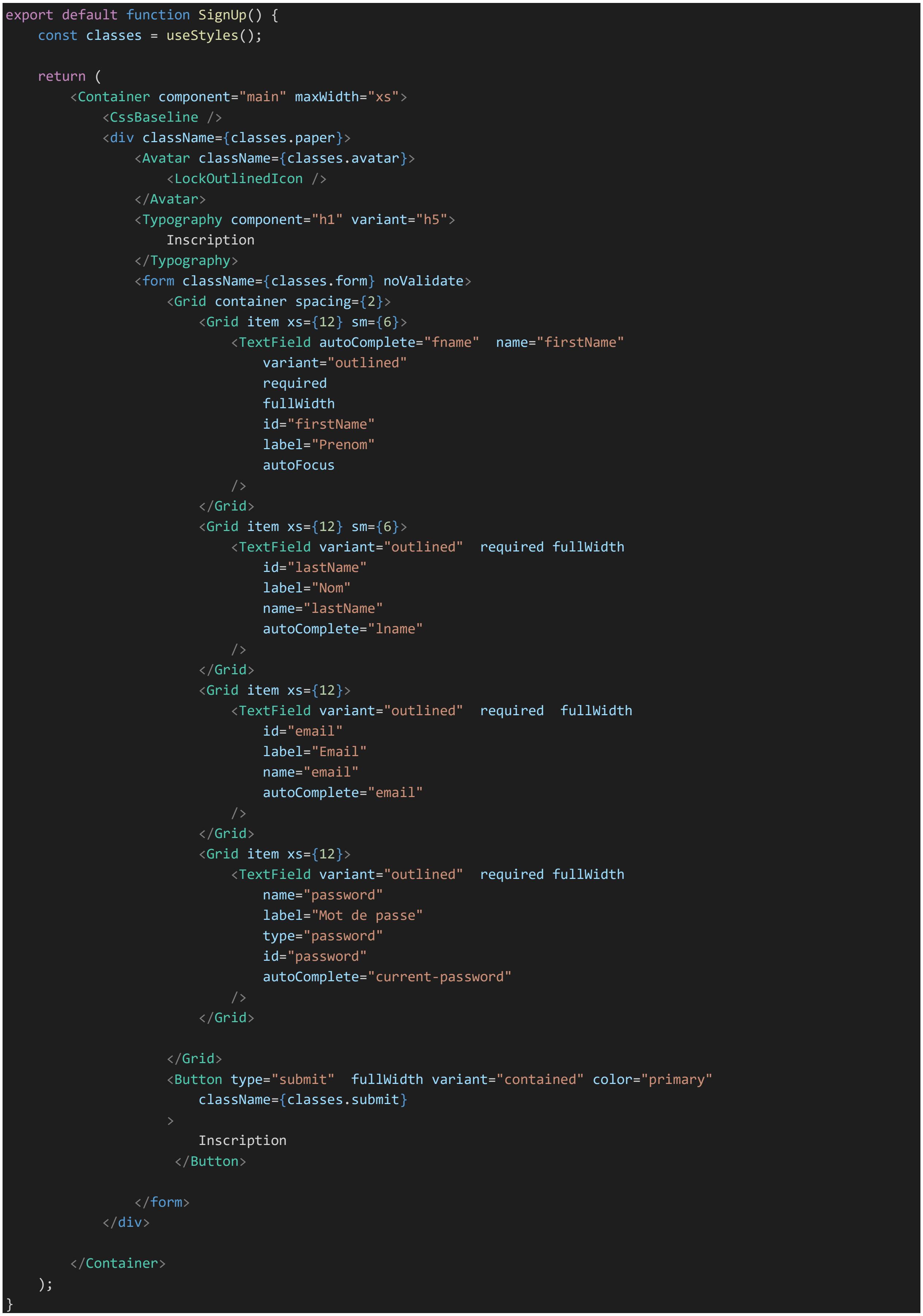
➢ Page Inscription:



Code :

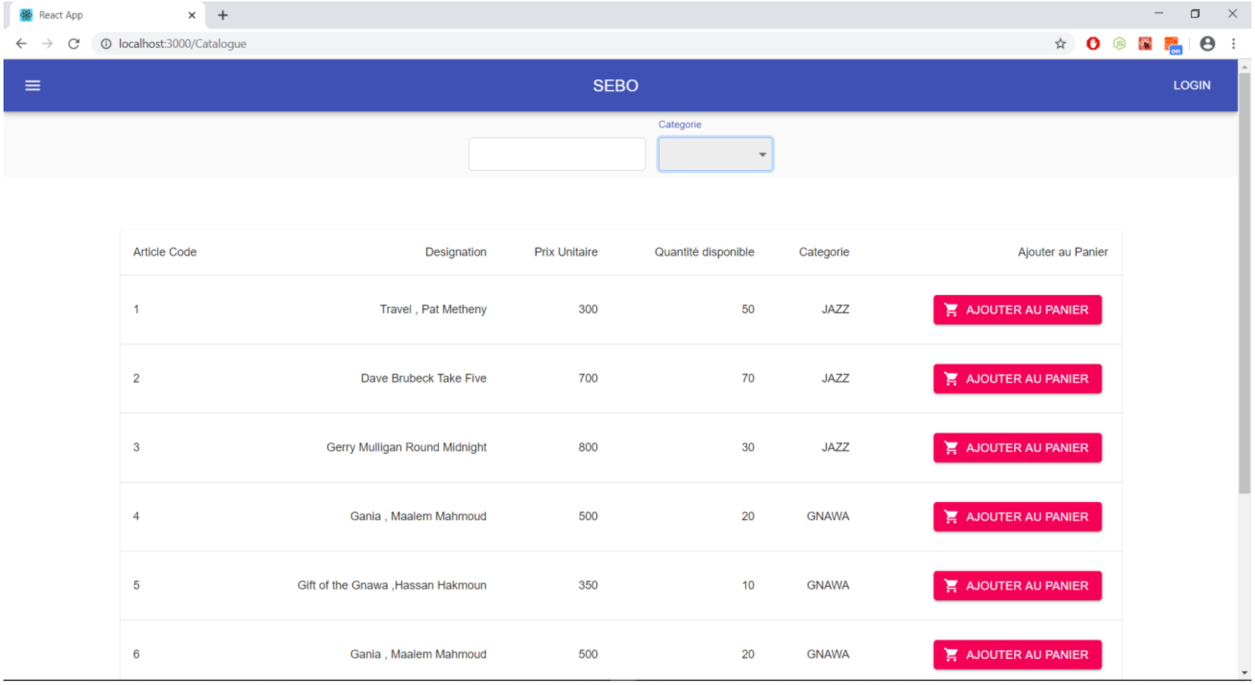


```
import React from 'react';
import Avatar from '@material-ui/core/Avatar';
import Button from '@material-ui/core/Button';
import CssBaseline from '@material-ui/core/CssBaseline';
import TextField from '@material-ui/core/TextField';
import Link from '@material-ui/core/Link';
import Grid from '@material-ui/core/Grid';
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(3),
    },
    submit: {
        margin: theme.spacing(3, 0, 2),
    },
}));
```



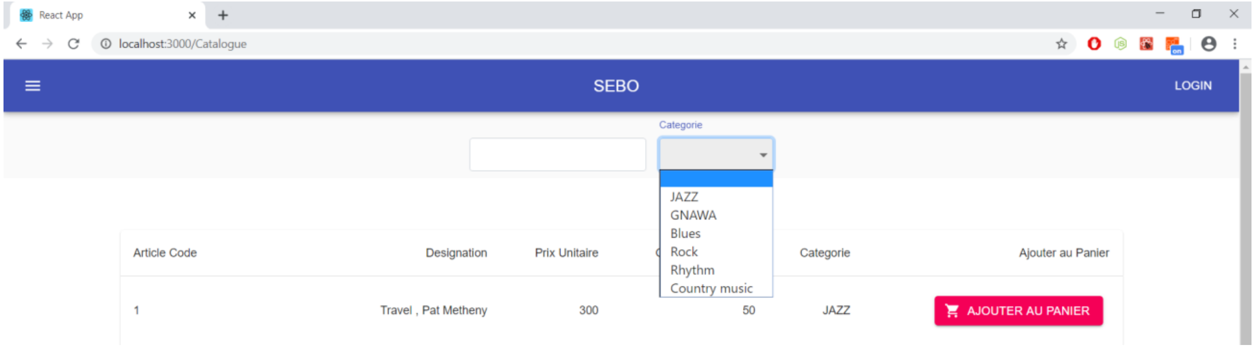
```
export default function SignUp() {
    const classes = useStyles();

    return (
        <Container component="main" 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
                            />
                        </Grid>
                        <Grid item xs={12} sm={6}>
                            <TextField variant="outlined"  required fullWidth
                                id="lastName"
                                label="Nom"
                                name="lastName"
                                autoComplete="lname"
                            />
                        </Grid>
                        <Grid item xs={12}>
                            <TextField variant="outlined"  required  fullWidth
                                id="email"
                                label="Email"
                                name="email"
                                autoComplete="email"
                            />
                        </Grid>
                        <Grid item xs={12}>
                            <TextField variant="outlined"  required fullWidth
                                name="password"
                                label="Mot de passe"
                                type="password"
                                id="password"
                                autoComplete="current-password"
                            />
                        </Grid>

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

                </form>
            </div>

        </Container>
    );
}
```

➢ Page Catalogue:



Article Code	Designation	Prix Unitaire	Quantité disponible	Categorie	Ajouter au Panier
1	Travel , Pat Metheny	300	50	JAZZ	AJOUTER AU PANIER
2	Dave Brubeck Take Five	700	70	JAZZ	AJOUTER AU PANIER
3	Gerry Mulligan Round Midnight	800	30	JAZZ	AJOUTER AU PANIER
4	Gania , Maalem Mahmoud	500	20	GNAWA	AJOUTER AU PANIER
5	Gift of the Gnawa ,Hassan Hakmoun	350	10	GNAWA	AJOUTER AU PANIER
6	Gania , Maalem Mahmoud	500	20	GNAWA	AJOUTER AU PANIER

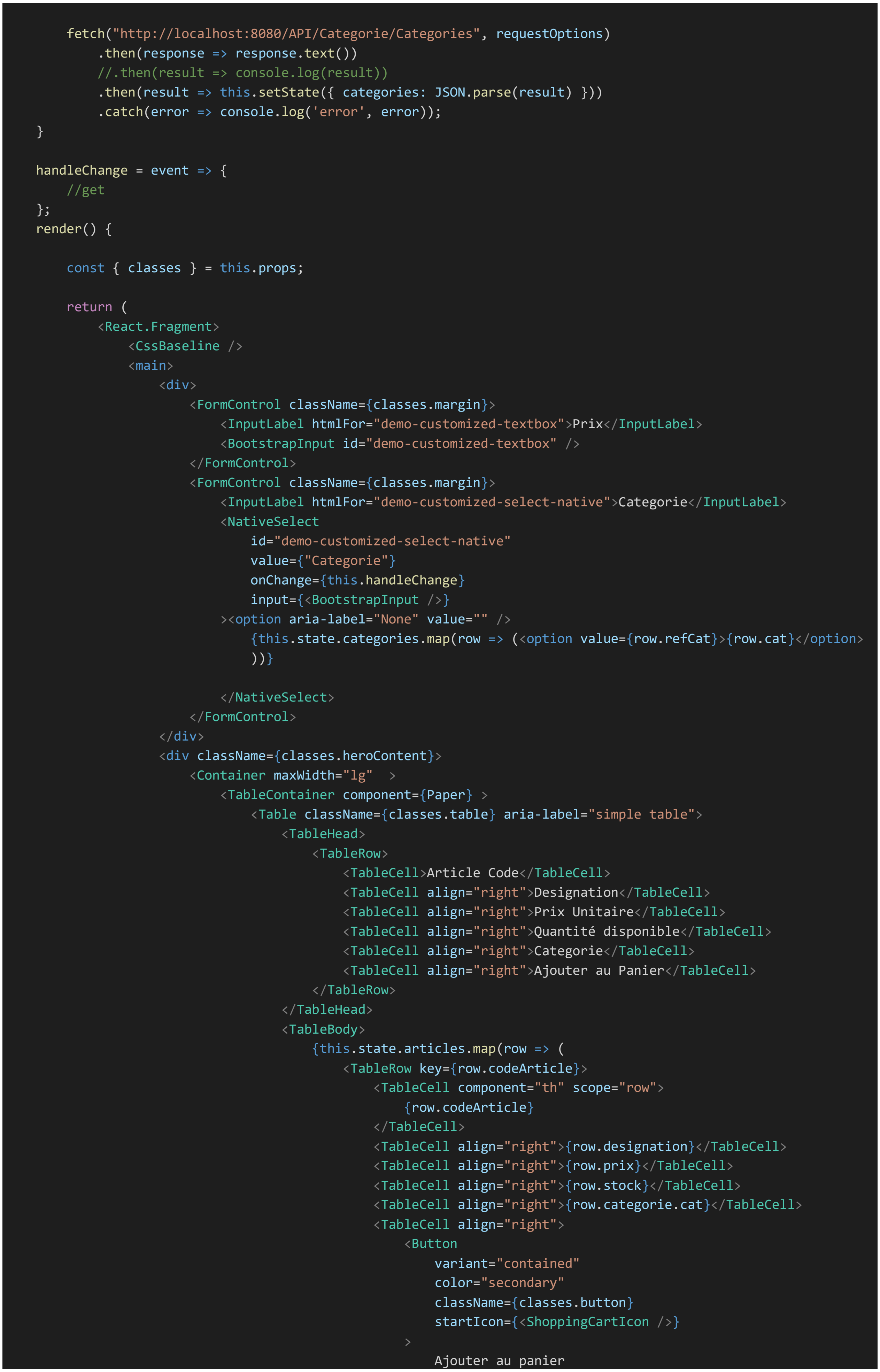
Filter par categorie



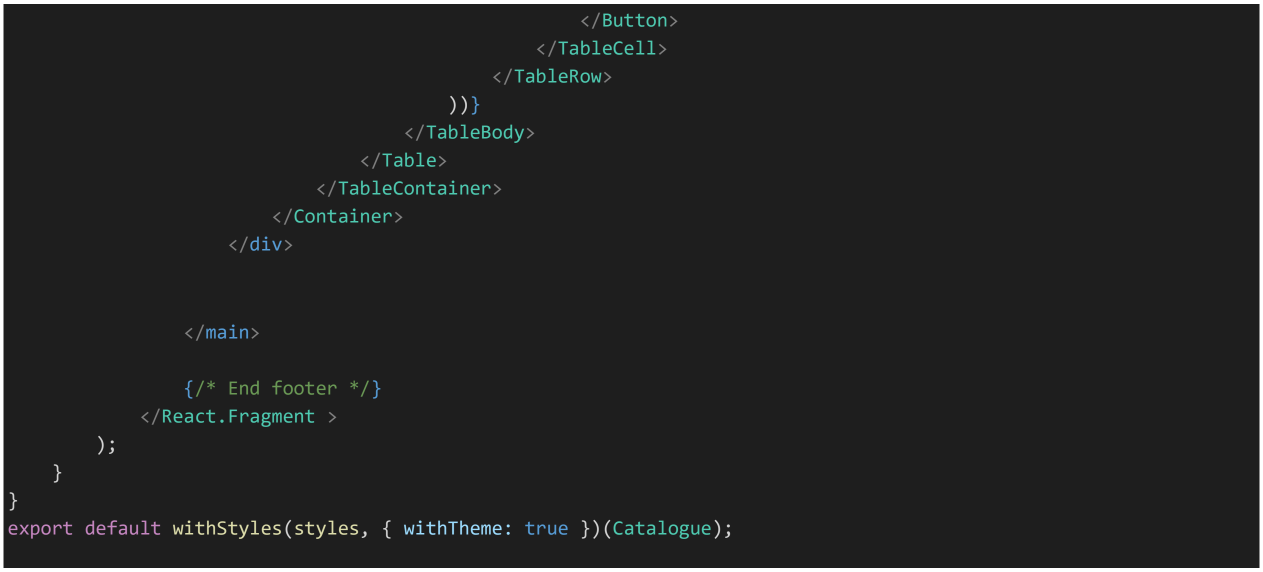
Article Code	Designation	Prix Unitaire		Categorie	Ajouter au Panier
1	Travel , Pat Metheny	300	50	JAZZ	AJOUTER AU PANIER

Code :

```
import React from "react";
import CssBaseline from '@material-ui/core/CssBaseline';
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 InputBase from '@material-ui/core/InputBase';
import InputLabel from '@material-ui/core/InputLabel';
import FormControl from '@material-ui/core/FormControl';
import NativeSelect from '@material-ui/core/NativeSelect';
import Button from '@material-ui/core/Button';
import ShoppingCartIcon from '@material-ui/icons/ShoppingCart';


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 BootstrapInput = withStyles(theme => ({
    root: {
        'label + &': {
            marginTop: theme.spacing(3),
        },
    },
    input: {
        borderRadius: 4,
        position: 'relative',
        backgroundColor: theme.palette.background.paper,
        border: '1px solid #ced4da',
        fontSize: 16,
        padding: '10px 26px 10px 12px',
        transition: theme.transitions.create(['border-color', 'box-shadow']),
        // Use the system font instead of the default Roboto font.
        fontFamily: [
            '-apple-system',
            'BlinkMacSystemFont',
            '"Segoe UI"',
            'Roboto',
            '"Helvetica Neue"',
            'Arial',
            'sans-serif',
            '"Apple Color Emoji"',
            '"Segoe UI Emoji"',
            '"Segoe UI Symbol"',
        ].join(','),
        '&:focus': {
            borderRadius: 4,
            borderColor: '#80bdff',
            boxShadow: '0 0 0 0.2rem rgba(0,123,255,.25)',
        },
    },
}))(InputBase);

class Catalogue extends React.Component {

    state = {
        articles: [],
        categories: []
    }

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

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



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

    handleChange = event => {
        //get
    };
    render() {

        const { classes } = this.props;

        return (
            <React.Fragment>
                <CssBaseline />
                <main>
                    <div>
                        <FormControl className={classes.margin}>
                            <InputLabel htmlFor="demo-customized-textbox">Prix</InputLabel>
                            <BootstrapInput id="demo-customized-textbox" />
                        </FormControl>
                        <FormControl className={classes.margin}>
                            <InputLabel htmlFor="demo-customized-select-native">Categorie</InputLabel>
                            <NativeSelect
                                id="demo-customized-select-native"
                                value={"Categorie"}
                                onChange={this.handleChange}
                                input={<BootstrapInput />}
                            ><option aria-label="None" value="" />
                                {this.state.categories.map(row => (<option value={row.refCat}>{row.cat}</option>
                                ))}

                            </NativeSelect>
                        </FormControl>
                    </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 align="right">Designation</TableCell>
                                            <TableCell align="right">Prix Unitaire</TableCell>
                                            <TableCell align="right">Quantité disponible</TableCell>
                                            <TableCell align="right">Categorie</TableCell>
                                            <TableCell align="right">Ajouter au Panier</TableCell>
                                        </TableRow>
                                    </TableHead>
                                    <TableBody>
                                        {this.state.articles.map(row => (
                                            <TableRow key={row.codeArticle}>
                                                <TableCell component="th" scope="row">
                                                    {row.codeArticle}
                                                </TableCell>
                                                <TableCell align="right">{row.designation}</TableCell>
                                                <TableCell align="right">{row.prix}</TableCell>
                                                <TableCell align="right">{row.stock}</TableCell>
                                                <TableCell align="right">{row.categorie.cat}</TableCell>
                                                <TableCell align="right">
                                                    <Button
                                                        variant="contained"
                                                        color="secondary"
                                                        className={classes.button}
                                                        startIcon={<ShoppingCartIcon />}
                                                    >
                                                        Ajouter au panier
```



```
                                                    </Button>
                                                </TableCell>
                                            </TableRow>
                                        ))}
                                    </TableBody>
                                </Table>
                            </TableContainer>
                        </Container>
                    </div>


                </main>

                {/* End footer */}
            </React.Fragment >
        );
    }
}
export default withStyles(styles, { withTheme: true })(Catalogue);
```