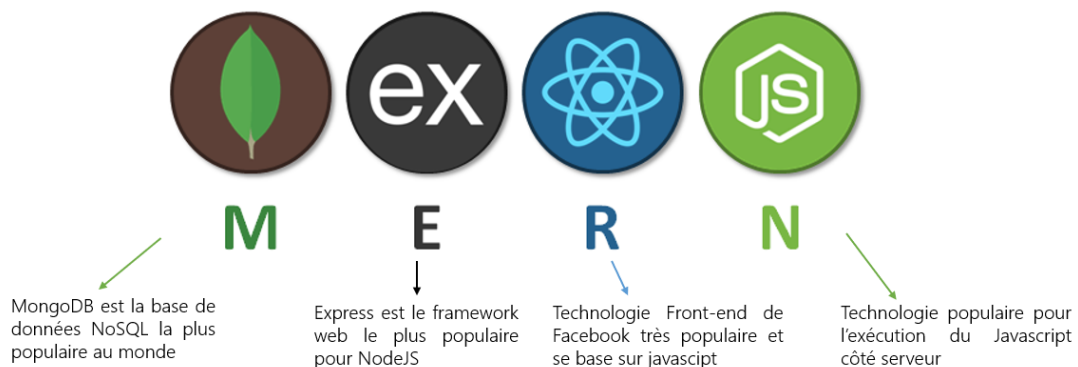


Création d'une application MERN – Partie 1 : Front end

MERN est une pile logicielle JavaScript gratuite et open source pour la création des applications Web dynamiques. MERN est l'acronyme de MongoDB, Express, React et Node.js. MERN est une technologie dérivée de la populaire pile MEAN (MongoDB, Express, Angular, Node) où, au lieu d'utiliser le Framework front-end Angular, nous utilisons React. Une autre variante populaire est le MEVN où nous utilisons Vue comme framework front-end.

Dans cette partie, nous allons aborder MERN à travers une application e-Commerce.



1. Création d'une application React

Pour créer rapidement une application react. Ouvrir le terminal et lancer la commande suivante:

`npx create-react-app frontend`

Cette commande créera le Template d'une application react très simple dans un dossier appelé frontend.

```
PS C:\Users\ACER\Desktop\Ecommerce> npx create-react-app frontend
Creating a new React app in C:\Users\ACER\Desktop\Ecommerce\frontend.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
[.....] \ idealTree:frontend: sill idealTree buildDeps
```

Si tout se passe bien, vous allez avoir la fenêtre suivante, taper les commandes suivantes pour accéder au dossier frontend et lancer l'application avec la commande `npm start`

```
We suggest that you begin by typing:

cd frontend
npm start

Happy hacking!
PS C:\Users\ACER\Desktop\Ecommerce> |
```

Lancer les deux commandes indiquées pour :

1. Accéder au dossier frontend
2. Lancer l'application en tapant `npm start`

Sur le terminal, vous allez avoir le résultat suivant :

```
Compiled successfully!

You can now view frontend in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.0.120:3000

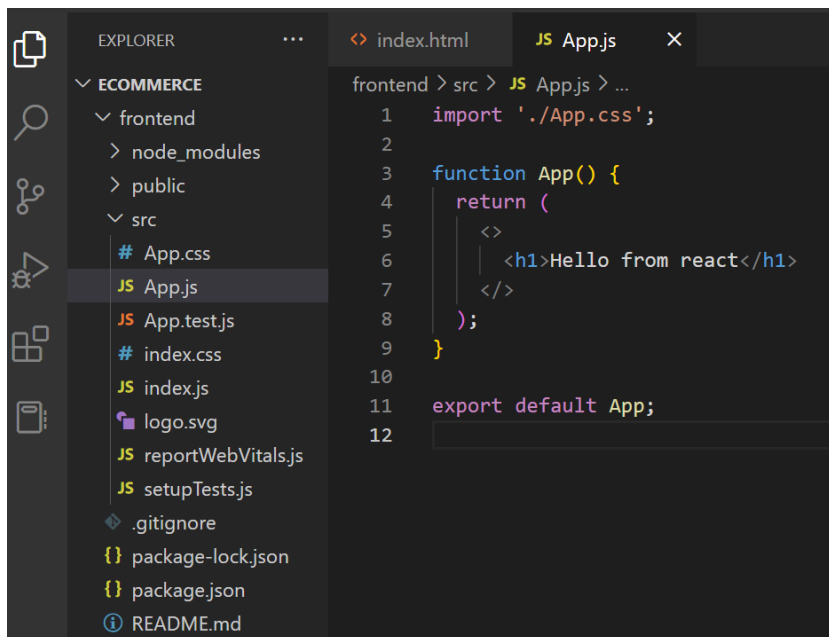
Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

En ouvrant l'url générée, vous obtiendrez un résultat qui ressemble à la figure suivante

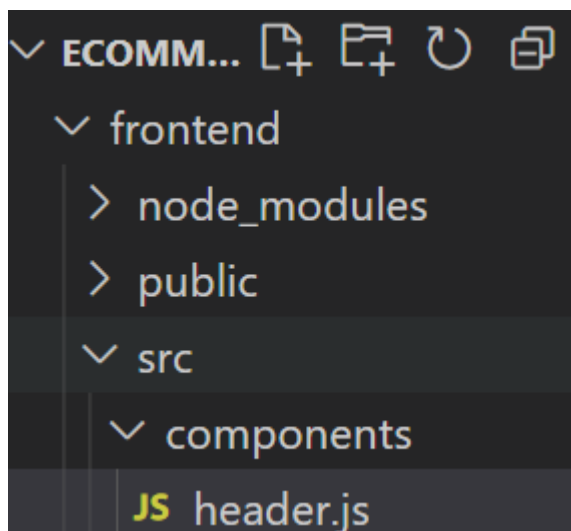


- Visualiser l'arborescence du projet. Ensuite, ouvrez le fichier appelé App.js comme indiqué sur la figure ci-dessous

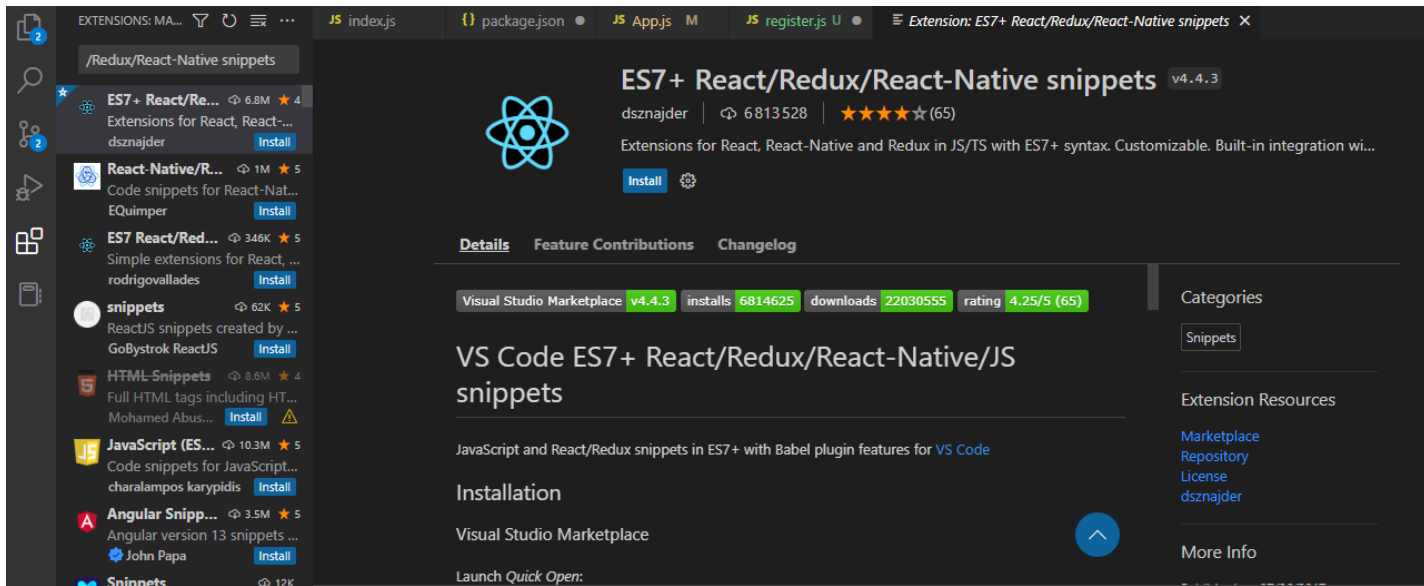


```
1 import './App.css';
2
3 function App() {
4   return (
5     <>
6     <h1>Hello from react</h1>
7     </>
8   );
9 }
10
11 export default App;
12
```

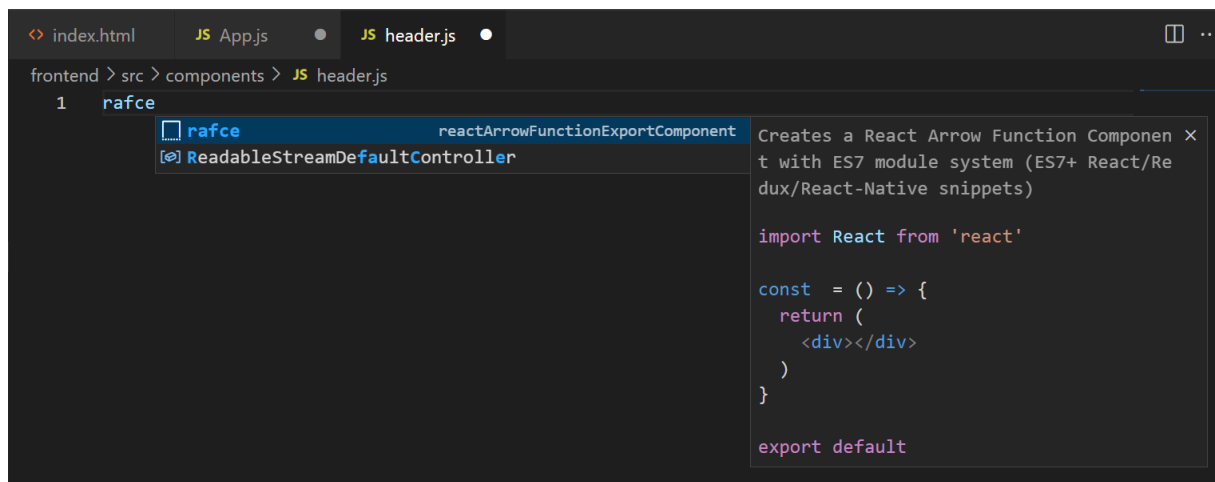
- Réaliser des modifications sur le fichier « App.js » et visualiser les résultats directement sur le navigateur.
- Dans le dossier src ajouter un sous dossier appelé components. Dans ce dossier, vous allez ajouter des composants (des parties) de votre application. Par exemple, nous commencerons par ajouter à ce dossier deux composants (fichiers javascript) appelés «header.js » et «footer.js».



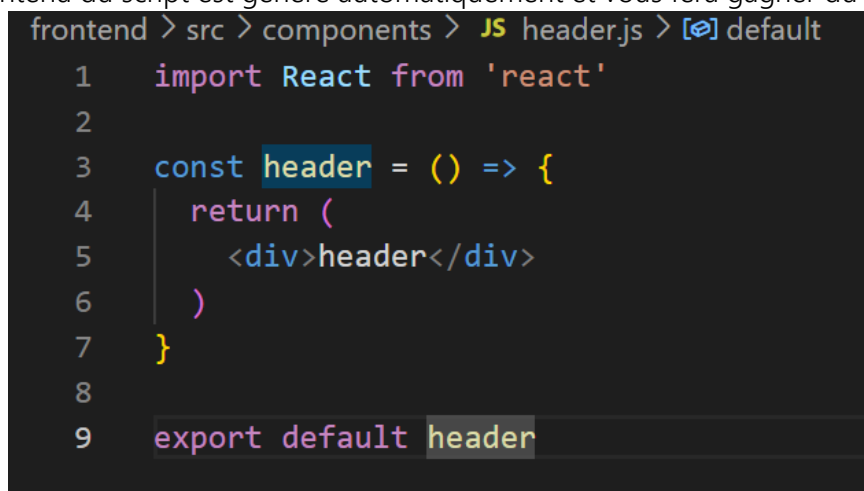
Pour générer du code rapidement dans react, il est recommandé d'installer l'extension *ES7+ React/Redux/React-Native snippets* sur VS Code.



Ouvrir le fichier header.js et taper *rafce* puis appuyer sur entrée comme le montre la figure suivante :



Le contenu du script est généré automatiquement et vous fera gagner du temps.



De la même façon ajouter un composant footer dans un script javascript que vous appellerez «footer.js »

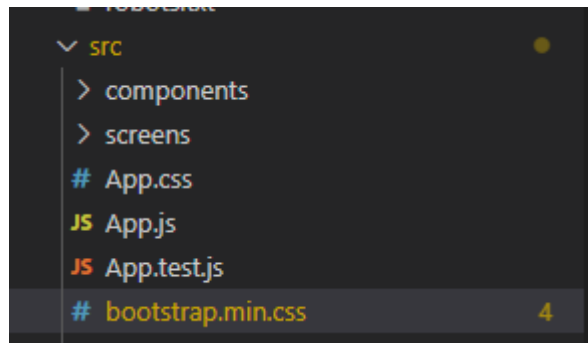
- Revenir au script App.js pour inclure les deux composants que vous venez de créer:

```
import './App.css';
import Header from './components/Header'
import Footer from './components/Footer'
const App=() => {
  return (
    <>
    <Header />
    <h1>Hello from react</h1>
    <Footer />
    </>
  );
}
```

2. Personnaliser l'interface graphique

Accéder au site web suivant pour télécharger très rapidement un thème Bootstrap : [Bootswatch: Free themes for Bootstrap](https://getbootstrap.com/docs/5.2/themes/)

Télécharger un thème (Fichier appelé bootstrap.min.css) et le mettre dans le dossier src comme le montre la figure suivante :



Ouvrir la page appelée «index.js» et ajouter une référence vers le fichier bootstrap.min.css

```
JS index.js  X
frontend > src > JS index.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import App from './App';
5  import './bootstrap.min.css'
6  import reportWebVitals from './reportWebVitals';
7
8  const root = ReactDOM.createRoot(document.getElementById('root'));
9  root.render(
10   <React.StrictMode>
11     <App />
12   </React.StrictMode>
13 );
```

- Exécuter l'application pour voir l'application du thème que vous avez choisi.
- 3. Installation du module Bootstrap
- Assurez-vous que vous êtes sur le dossier frontend, sinon accéder au dossier avec la commande `cd`
- Installer le module react-bootstrap via la commande `npm -i`

```
PS C:\Users\ACER\Desktop\Ecommerce> cd frontend
PS C:\Users\ACER\Desktop\Ecommerce\frontend> npm i react-bootstrap
```

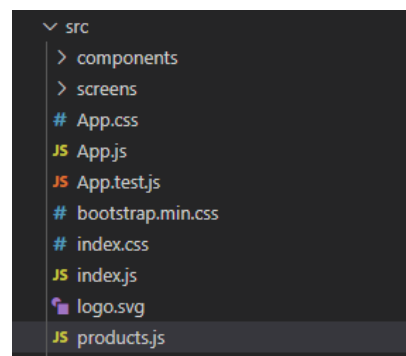
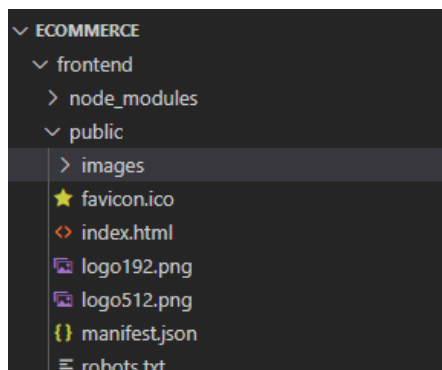
Pour ajouter des composants rapidement à votre interface graphique, accéder au site web suivant :

[React-Bootstrap · React-Bootstrap Documentation](#)

4. Ajouter des composants Bootstrap

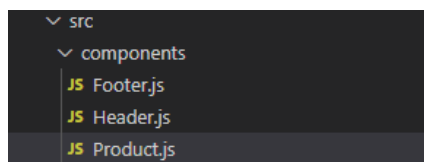
Ajouter le dossier images qui vous a été fourni au dossier public comme suit :

Copier le fichier appelé `products.js` dans le dossier appelé `src` de votre projet



- Créer un nouveau composant dans le dossier appelé «components» appelé `Product.js`

Création d'un fichier javascript appelé «Product.js» qui va contenir l'affichage de chaque produit



```
import React from 'react'
import Card from 'react-bootstrap/Card';
import Button from 'react-bootstrap/Button';
const Product = ({product}) => {
  return (
    <Card className='my-3 p-3 rounded'>
      <Card.Img variant="top" src={product.image} />
      <Card.Body>
        <Card.Title>{product.name}</Card.Title>
        <Card.Text>
          Some quick example text to build on the card title
          and make up the bulk of the card's content.
        </Card.Text>
        <Button variant="primary">Go somewhere</Button>
      </Card.Body>
    </Card>
  )
}
export default Product
```

Créer un nouveau composant appelé ProductList.js dont le contenu est comme suit:

```
import React from 'react'
import {Row, Col} from 'react-bootstrap'
import products from '../products.js'
import Product from './Product'
import Pagination from 'react-bootstrap/Pagination';

const ProductList = () => {
  return (
    <>
    <Row>
    {products.map((product) => (
      <Col key={product._id} sm={12} md={6} lg={4} xl={3}>
        <Product product={product} />
      </Col>
    ))}
    </Row>
  );
  </>
)
}

export default ProductList
```