

React.js with PHP API REST

5. Composants Génériques et props children

Après avoir développé les pages SeriesScreen et TechsScreen et les composants SerieCard et TechCard nous constatons des similitudes dans le code. Nous allons donc créer un premier composant générique Card qui reprendra le code commun à nos 3 cards (Article, Serie et Tech) Nous créons le composant dans le répertoire src/components/cards

```
src > components > cards > Card.jsx > Card
1  import PropTypes from 'prop-types';
2  function Card({children, title, imgSrc, link, btnText}) {
3      return (
4          <div className="c col-md-6 col-lg-3 d-flex align-items-stretch justify-content-center">
5              <div className="card mb-3" style={{width: "18rem"}}>
6                  <img src={imgSrc} className="card-img-top"
7                      alt={title} />
8                  <div className="card-body text-center">
9                      <h5 className="card-title">{title}</h5>
10                     {children}
11                     <a href={link} className="btn btn-primary">{btnText}</a>
12                 </div>
13             </div>
14         </div>
15     );
16 }
17 Card.propTypes = {
18     children: PropTypes.oneOfType([
19         PropTypes.arrayOf(PropTypes.element),
20         PropTypes.element
21     ]),
22     title: PropTypes.string.isRequired,
23     imgSrc: PropTypes.string.isRequired,
24     link: PropTypes.string.isRequired,
25     btnText: PropTypes.string.isRequired,
26 }
27 export default Card;
```

children est une props spéciale prédéfinie en React.js qui permet de passer des éléments enfants dans un composant. Elle est utilisée pour rendre le contenu imbriqué dans un composant et permet de créer des composants plus flexibles et réutilisables.

Ressources :

<https://react.dev/learn/passing-props-to-a-component#passing-jsx-as-children>

Nous pouvons maintenant nous servir de ce composant pour développer les composants ArticleCard, SerieCard et TechCard

Nous déplaçons ArticleCard dans le répertoire src/components/cards et le modifions pour utiliser Card

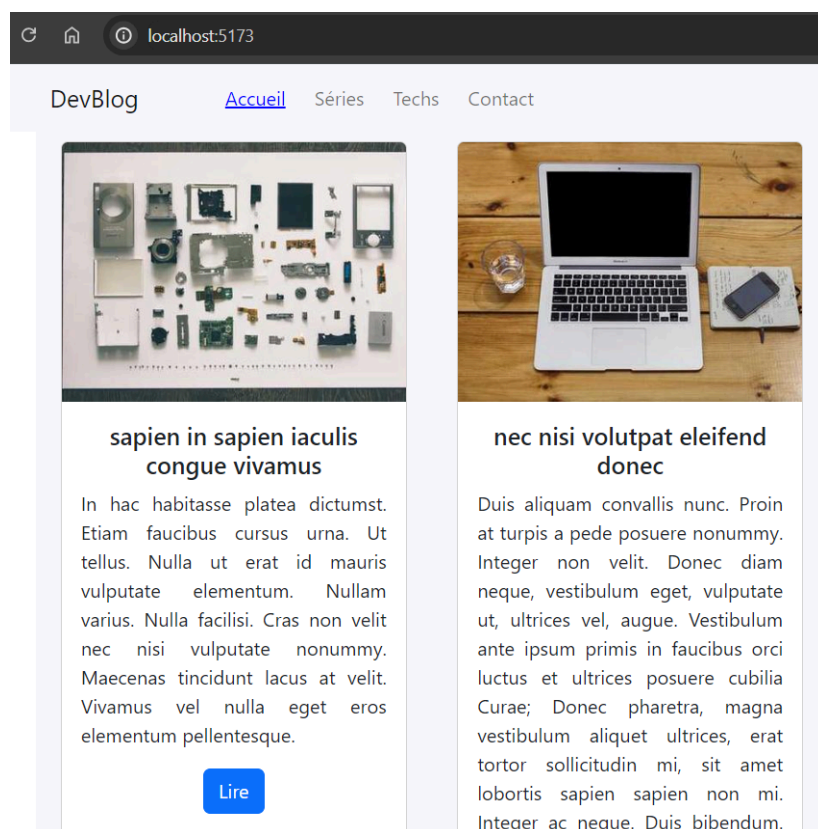
```

src > components > cards > ArticleCard.jsx > ...
1  import PropTypes from "prop-types";
2  import Card from "../Card";
3  function ArticleCard(props) {
4      const { article } = props;
5      return (
6          <Card
7              title={article.title}
8              imgSrc={article.img_src}
9              link={"articles/details/" + article.id_article}
10             btnText="Lire"
11         >
12             <p className="card-text" style={{ textAlign: "justify" }}>
13                 {article.summary}
14             </p>
15         </Card>
16     );
17 }
18 ArticleCard.propTypes = {

```

Le paragraphe (ligne 12 à 14) remplacera children dans Card

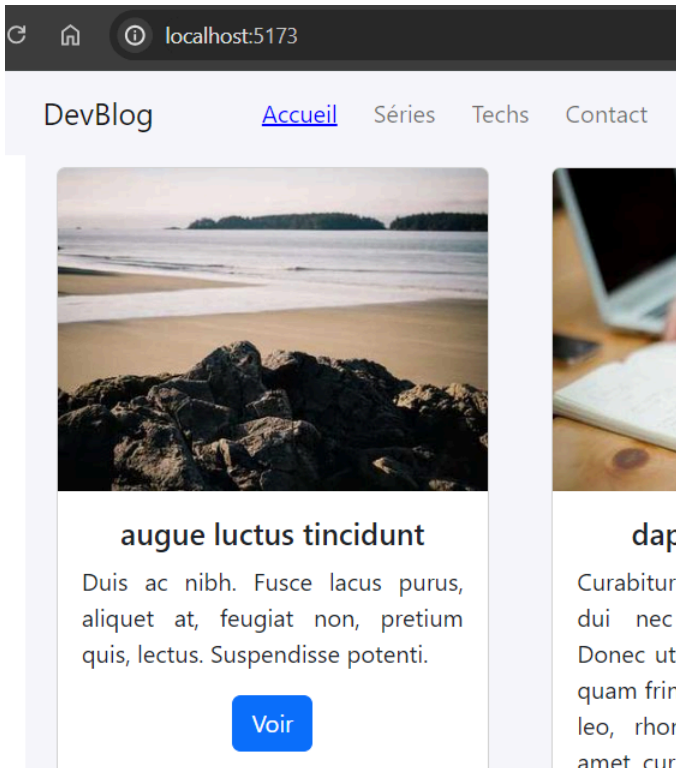
Résultat dans HomeScreen



Même principe pour SerieCard

```
src > components > cards > SerieCard.jsx > SerieCard
1  import PropTypes from "prop-types";
2  import Card from "../Card";
3  function SerieCard(props) {
4    const { serie } = props;
5    return (
6      <Card
7        title={serie.title}
8        imgSrc={serie.img_src}
9        link={"series/details/" + serie.id_serie}
10       btnText="Voir"
11     >
12       <p className="card-text" style={{ textAlign: "justify" }}>
13         {serie.summary}
14       </p>
15     </Card>
16   );
17 }
18 SerieCard.propTypes = {
19   serie: PropTypes.shape({
20     id_serie: PropTypes.number.isRequired,
21     title: PropTypes.string,
22     summary: PropTypes.string,
23     img_src: PropTypes.string,
24     is_deleted: PropTypes.bool,
25   }),
26 };
27 export default SerieCard;
```

Résultat sur SeriesScreen

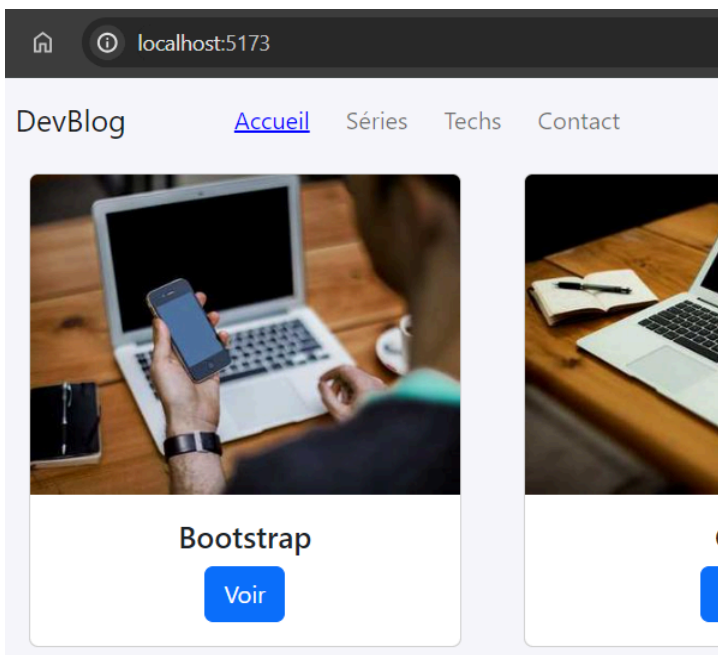


TechCard

```
src > components > cards > TechCard.jsx > TechCard
1  import PropTypes from "prop-types";
2  import Card from "../Card";
3  function TechCard(props) {
4      const { tech } = props;
5      return (
6          <Card
7              title={tech.label}
8              imgSrc={tech.img_src}
9              link={"techs/details/" + tech.id_tech}
10             btnText="Voir"
11         />
12     );
13 }
14 TechCard.propTypes = {
15     tech: PropTypes.shape({
16         id_tech: PropTypes.number.isRequired,
17         label: PropTypes.string,
18         img_src: PropTypes.string,
19         is_deleted: PropTypes.bool,
20     }),
21 };
22 export default TechCard;
```

Cette fois ci pas de children passé à Card

Résultat sur TechsScreen



A vous de faire la même chose pour les 3 screens (Home, Series et Techs) qui présentent des similitudes en créant un composant ListScreen.jsx

git : https://github.com/DWWM-23526/REACT_BLOG_PROJECT/tree/Step05