

Atelier n°2 Partie 2 :

Composants fonctionnels - Routage

Objectifs :

- Appliquer le routage à notre application React.
- Ajouter le Lazy loading aux composants existants.

Travail à faire :

Partie 2 : Routage

- 1) Ajoutez à votre projet package **react-router-dom**.

Remarque : pour ajouter un package à votre projet vous pouvez utiliser le gestionnaire de paquets yarn ou npm.

**npm install react-router-dom
ou yarn add react-router-dom**

- 2) Importer à votre index.js le composant BrowserRouter qui appartient au package react-router-dom.
- 3) Mettre à jour le code index.js pour configurer le routage.
- 4) Importer à votre App.js les composants Route et Routes qui appartiennent au package react-router-dom.
- 5) Mettre à jour le code App.js avec le routage spécifique.

Remarque :

- Vous allez créer un nouveau composant fonctionnel sous le dossier **Components** appelé **NotFound** qui sera rendu, si l'utilisateur donne une route non déclarée.
- Vous allez utiliser l'image **notfound.jfif**.
- Vous allez utiliser les routes imbriquées.

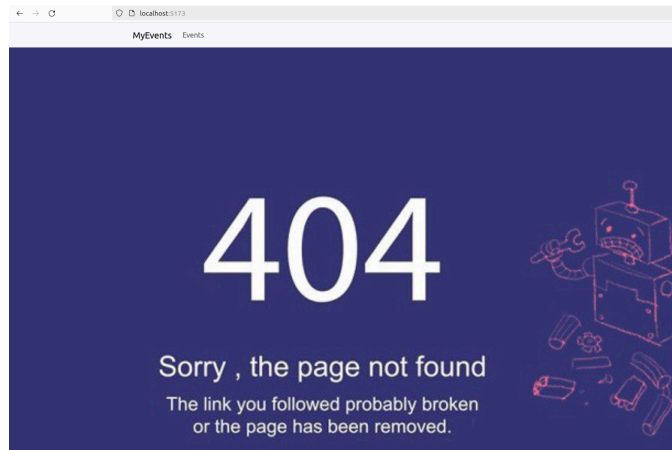


Figure 1 : “404 not found”

- 6) Créer un nouveau composant fonctionnel nommé **NavigationBar** dans un fichier **Navbar.js** sous le dossier **Components** qui sera la barre de navigation de la boutique.

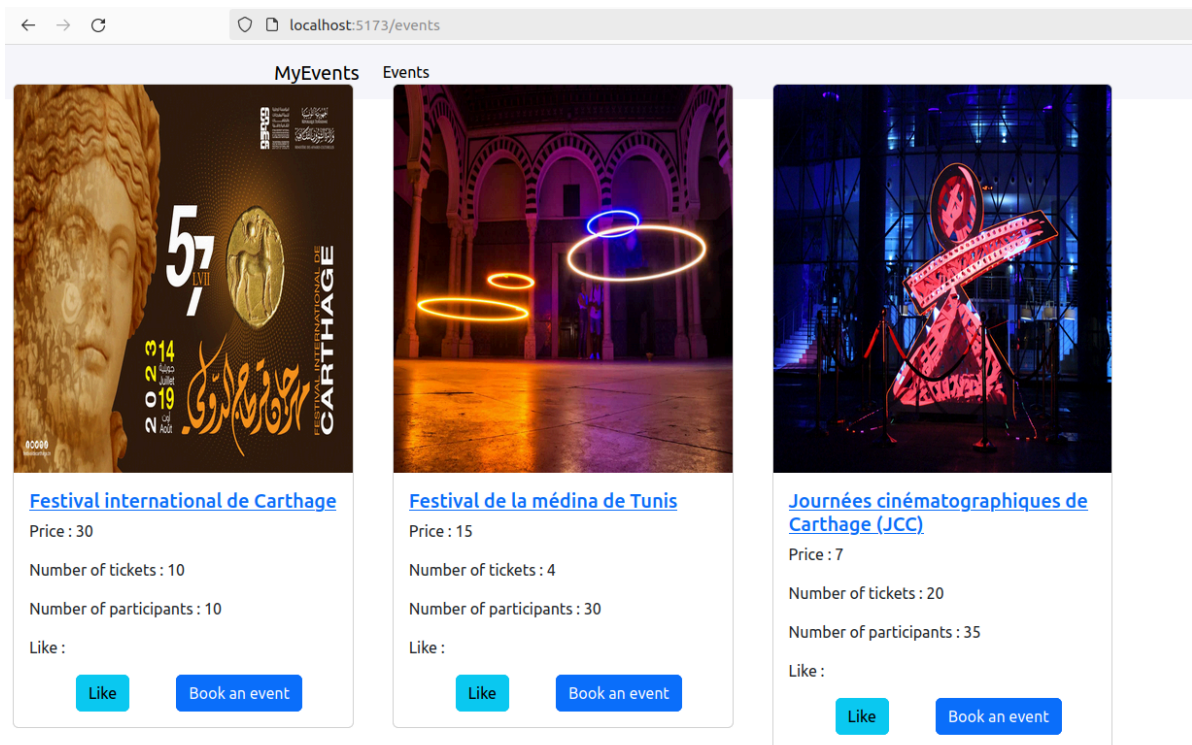


Figure 2 : “Navigation Bar”

Remarque :

- Vous allez utiliser le composant **NavBar** à partir de react-bootstrap.
 - Vous allez utiliser le composant **NavLink** et vérifier si la route est active ou non en soulignant la route active.
- 7) Mettre à jour le code **Event.js** pour qu'en cliquant sur le nom d'un événement les détails de l'événement sélectionné seront affichés.

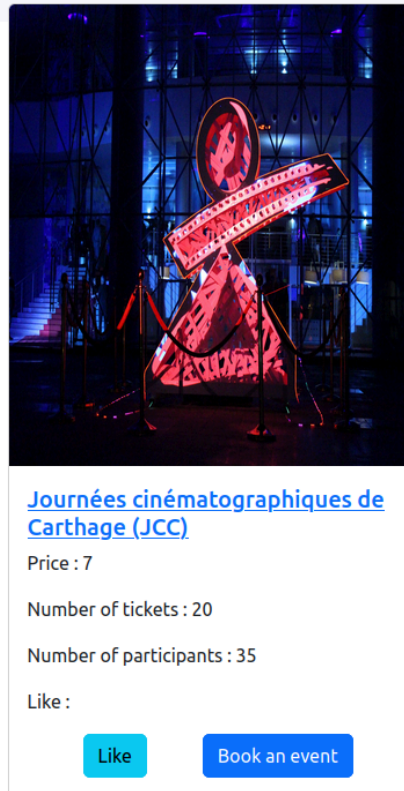


Figure 3 : “NavLink”

Remarque : vous allez créer un nouveau composant fonctionnel sous le dossier **Components** appelé **EventDetails** et utiliser le nom d’un événement pour le récupérer.



Figure 4 : “Event Details”

8) Appliquer le Lazy Loading qui appartient au package react pour votre routage.