

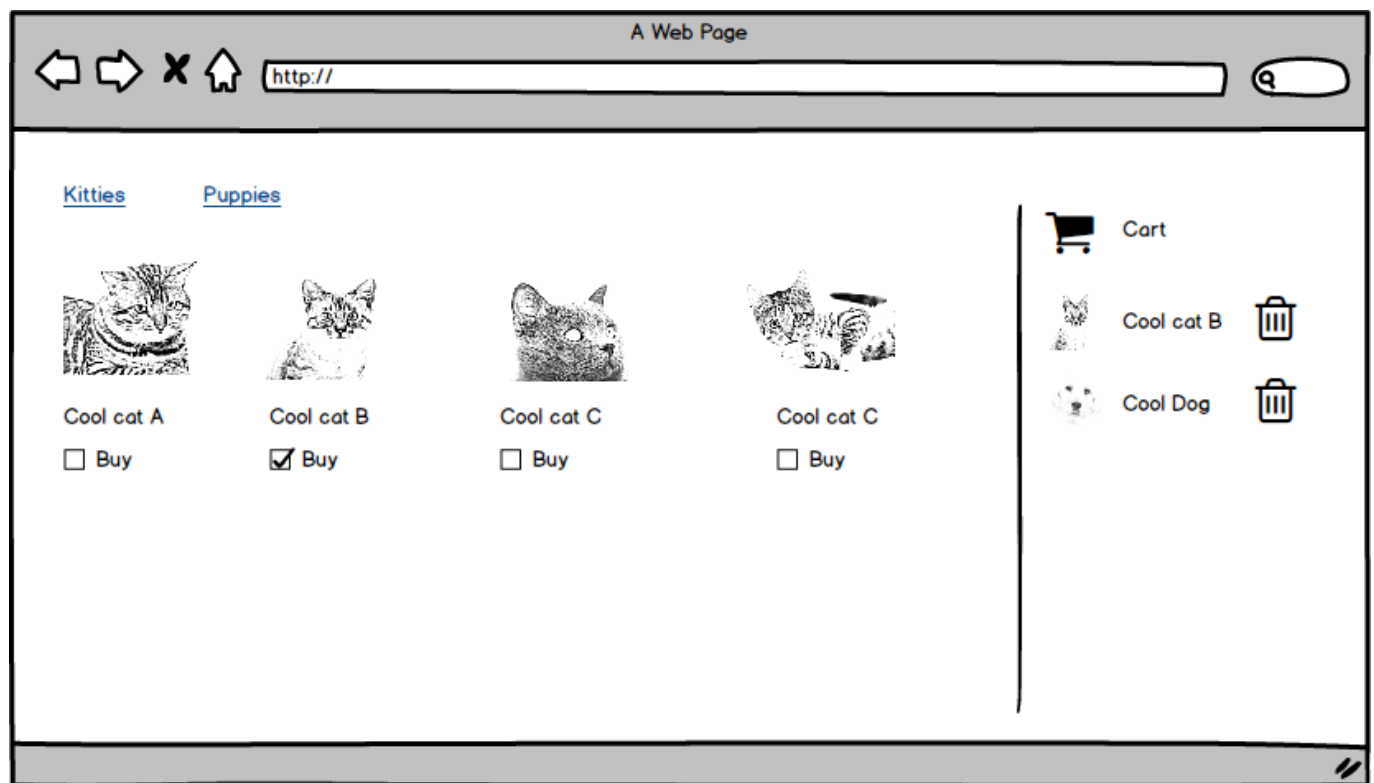
Listado imágenes (valido para entrega)

En este ejercicio vamos practicar con React route y el contexto.

Enunciado

Como miembro de un portal de un banco de imagenes nos piden tener un carrito de la compra siempre visible: - El usuario tiene una area principal en la que se muestra una página con imagenes, cada imagen tiene un checkbox que permite seleccionarlo y añadirlo al carrito de la compra (cada imagen tiene un ID).

- Para simplificar vamos a tener dos páginas distintas con imagenes seleccionables - Al derecha (fuera de React-router) mostramos un carrito de la compra.
- Cada vez que el usuario selecciona una imagen en una de las páginas se añade al carrito. - Si un usuario borra la imagen del carrito se deselecciona de la página (en caso de que esté activa).



Pistas:

- Arranca por montar las páginas y que tenga unos datos mockeados con el siguiente interfaz para cada imagen:

```
interface PictureInfo {  
  id: string;  
  picUrl: string;  
  title: string;  
}
```

- Create un componente carrito y muéstralo fuera del routing (o si quieres podría definir un layout para esto)
- Una vez funcionando, piensa que para el carrito de la compra sólo te hace falta una lista de Id's de imágenes:
 - Arranca vacía.
 - Cada vez que selecciono una imagen la añado a ese array del contexto.
 - Si la eliminas del arrea del carrito se tiene que deseleccionar de la ventana que se este mostrando.
 - ¿Como hacemos esto?
 - Al cargar una página create un ViewModel con el siguiente campo de más:

```
interface PictureInfo {
    id : string;
    picUrl: string;
    title: string;
+   selected : boolean;
}
```

- Nada más cargar la página mira el contexto y pon a true los selected que toquen.
- Si marco en la página pongo el campo a true/false, y además hago un Set al contexto para actualizarlo (automáticamente se actualizará en el carrito).
- En el carrito iría de la otra manera, si deschequeo el carrito hago un check y la ventana de turno estar usando un *useEffect* escuchando al ventana del carrito para actualizar sus datos.

Ideas opcionales:

- Haz que el carrito se puede ocultar.
- Implementa una página de "checkout" para hacer el pedido.
- Estilalo con material UI.
- Añade una opción de vaciar carrito.
- ...