



# **</> alyra-shopping-list**

# Contents

## ■ alyra-shopping-list

📄 package.json

### ■ public

📄 index.html

📄 manifest.json

📄 robots.txt

### ■ src

📄 App.css

📄 App.js

📄 App.test.js

📄 index.css

📄 index.html

📄 index.js

📄 reportWebVitals.js

📄 setupTests.js

### ■ components

📄 AddPopularProduct.js

📄 AddPopularProductForm.js

📄 Header.js

📄 Product.js

📄 ShoppingApp.js

📄 ShoppingList.js

# package.json

/package.json

```
{
  "name": "alyra-shopping-list",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@fontsource/kiwi-maru": "^4.2.1",
    "@testing-library/jest-dom": "^5.11.4",
    "@testing-library/react": "^11.1.0",
    "@testing-library/user-event": "^12.1.10",
    "bootstrap": "^5.0.0-beta3",
    "react": "^17.0.2",
    "react-dom": "^17.0.2",
    "react-scripts": "4.0.3",
    "web-vitals": "^1.0.1"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

# index.html

/public/index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See https://developers.google.com/web/
fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->
  </body>
</html>
```

# manifest.json

/public/manifest.json

```
{
  "short_name": "React App",
  "name": "Create React App Sample",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "64x64 32x32 24x24 16x16",
      "type": "image/x-icon"
    },
    {
      "src": "logo192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "logo512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff"
}
```

# robots.txt

/public/robots.txt
--------------------

# <https://www.robotstxt.org/robotstxt.html>  
User-agent: \*  
Disallow:

# App.css

/src/App.css

```
.App {
  text-align: center;
}

.App-logo {
  height: 40vmin;
  pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

.App-link {
  color: #61dafb;
}

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}
```

# App.js

/src/App.js

```
import React from 'react'
// import logo from './logo.svg'
import './App.css'
import Header from './components/Header'
import ShoppingApp from './components/ShoppingApp'

function App() {
  return (
    <div className="container">
      <Header />
      <ShoppingApp />
    </div>
  )
}

export default App
```



## App.test.js

/src/App.test.js

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

## index.css

/src/index.css

```
body {  
  font-family: 'Kiwi Maru', sans-serif !important;  
}  
  
code {  
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',  
    monospace;  
}
```

# index.html

/src/index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See https://developers.google.com/web/
fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <link rel="stylesheet" href="./index.css" />

    <title>Alyra Shopping List</title>
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
```

# index.js

/src/index.js

```
import React from 'react'
import ReactDOM from 'react-dom'
import './index.css'
import App from './App'
import 'bootstrap/dist/css/bootstrap.css'
import '@fontsource/kiwi-maru'
import './index.css'

import reportWebVitals from './reportWebVitals'
// import '@fontsource/kiwi-maru'

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
)

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals()
```

# reportWebVitals.js

/src/reportWebVitals.js

```
const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) =>
    {
      getCLS(onPerfEntry);
      getFID(onPerfEntry);
      getFCP(onPerfEntry);
      getLCP(onPerfEntry);
      getTTFB(onPerfEntry);
    });
  }
};

export default reportWebVitals;
```

## setupTests.js

/src/setupTests.js

```
// jest-dom adds custom jest matchers for asserting on DOM nodes.  
// allows you to do things like:  
// expect(element).toHaveTextContent(/react/i)  
// learn more: https://github.com/testing-library/jest-dom  
import '@testing-library/jest-dom';
```

# AddPopularProduct.js

```
/src/components/AddPopularProduct.js
```

```
import React from 'react'
import AddPopularProductForm from './AddPopularProductForm'

const AddPopularProduct = (props) => {
  const { shopList, addToShopList } = props
  const populars = [
    { text: 'pain', emoji: '?' },
    { text: 'lait', emoji: '?' },
    { text: 'pizza', emoji: '?' },
    { text: 'salade', emoji: '?' },
    { text: 'oranges', emoji: '?' },
  ]

  return (
    <section>
      <div className="bg-light border p-4">
        <h2 class="mb-3 h4">Ajouter un produit :</h2>

        <AddPopularProductForm
          shopList={shopList}
          addToShopList={addToShopList}
        />

        <h3 className="h5">Avez vous besoin de ?</h3>
        <div className="mb-3 d-flex flex-wrap align-items-center">
          {populars.map((el) => (
            <button
              key={el.text}
              className="btn btn-outline-success me-2 mb-2 d-
flex align-items-center"
              onClick={() => addToShopList(el.text)}
              disabled={shopList.includes(el.text)}
            >
              {el.text}{' '}
              <span className="fs-1" role="img" aria-hidden>
                {el.emoji}
              </span>
            </button>
          ))}
        </div>
      </div>
    </section>
  )
}
export default AddPopularProduct
```

## AddPopularProductForm.js

/src/components/AddPopularProductForm.js

```
import React from 'react'
const AddPopularProductForm = (props) => {
  const { shopList, addToShopList } = props

  const handleFormSubmit = (event) => {
    // nous devons empêcher l'action par défaut de notre formulaire
    event.preventDefault()
    // récupérer la valeur depuis le champ input#product
    const newProduct = event.target.elements.product.value
    // s'assurer que la liste contient des produits uniques
    console.log('shopList:', shopList)
    console.log('newProduct:', newProduct)

    if (shopList.includes(newProduct)) {
      alert(`${newProduct} est déjà sur la liste`)
    } else {
      addToShopList(newProduct)
    }
    // vider l'input (remettre le formulaire à zéro)
    event.target.reset()
  }

  return (
    <form className="mb-5" onSubmit={handleFormSubmit}>
      <div className="input-group mb-2">
        <input
          id="product"
          className="form-control"
          aria-label="Ajouter sur la liste"
          required
        />
        <button type="submit" className="btn btn-success btn-lg">
          J'ajoute !
        </button>
      </div>
    </form>
  )
}

export default AddPopularProductForm
```



# Header.js

/src/components/Header.js

```
import React from 'react'
const Header = () => {
  return (
    <header className="text-center my-5">
      <h1>Ma liste des courses</h1>
      <p lang="en">Let's go shopping! Yay !!</p>
    </header>
  )
}

export default Header
```

# Product.js

/src/components/Product.js

```
import React from 'react'
const Product = (props) => {
  const { product, removeFromShopList } = props

  const handleButtonClick = () => {
    removeFromShopList(product)
  }

  return (
    <div className="d-flex align-items-center justify-content-between">
      {product}
      <button
        className="btn btn-sm btn-warning"
        onClick={handleButtonClick}
      >
        <span role="img" aria-hidden>
          ✖?
        </span>{' '}
        ok
      </button>
    </div>
  )
}

export default Product
```

# ShoppingApp.js

/src/components/ShoppingApp.js

```
import React from 'react'
import { useState } from 'react'
import ShoppingList from '../ShoppingList'
import AddPopularProduct from '../AddPopularProduct'

const ShoppingApp = () => {
  const [shopList, setShopList] = useState([])

  const addToShopList = (product) => {
    setShopList([...shopList, product])
  }

  const removeFromShopList = (product) => {
    setShopList(shopList.filter((el) => el !== product))
  }

  return (
    <main className="row">
      <section className="col-lg-8">
        <ShoppingList
          shopList={shopList}
          removeFromShopList={removeFromShopList}
          addToShopList={addToShopList}
        />
      </section>
      <section className="col-lg-4">
        <AddPopularProduct
          shopList={shopList}
          addToShopList={addToShopList}
        />
      </section>
    </main>
  )
}

export default ShoppingApp
```

# ShoppingList.js

```
/src/components/ShoppingList.js
```

```
import React from 'react'
import Product from '../Product'
import { useState } from 'react'

const ShoppingList = (props) => {
  const { shopList, removeFromShopList } = props

  const [filter, setFilter] = useState('')

  const filteredList = shopList.filter((el) =>
    el.trim().toLowerCase().startsWith(filter.trim().toLowerCase())
  )
  return (
    <div>
      <h2 className="mb-3 h4">Produits à acheter
        ({shopList.length}):</h2>
      <div className="input-group mb-3">
        <span>
          role="img"
          aria-label="search"
          className="input-group-text"
        >
          ?
        </span>
        <input
          type="search"
          value={filter}
          onChange={(e) => setFilter(e.target.value)}
          placeholder="Rechercher dans votre liste des courses ..."
          aria-label="Chercher"
          className="form-control"
        />
      </div>
      {filter && (
        <p className="d-flex justify-content-between">
          <span>
            Produits qui commencent par <i>{filter}</i>
          </span>
          <button
            className="btn btn-light btn-sm"
            onClick={() => setFilter('')}
          >
            <span role="img" aria-hidden>
              ?
            </span>{' '}
            Réinitialiser
          </button>
        </p>
      )}
    </div>
  )
}
```

```

    })
    <ol className="list-group mb-3 shadow">
      {filteredList.map((el) => {
        return (
          <li className="list-group-item" key={el}>
            <Product
              product={el}
              removeFromShopList={removeFromShopList}
            />
          </li>
        )
      })}
    </ol>
  </>
)
}

export default ShoppingList

```