Creating UI for Cake types:

App.js:

```
import './App.css';
import {useState} from 'react';

function App() {
  return (
    <div className="App">
      <h1>Cakes</h1>
      <div>
        <h3>Chocolate Cake</h3>
        <h4>Rs 500</h4>
        <img src=" " />
        <button>Add to Cart</button>
      </div>

      <div>
        <h3>Butter Cake</h3>
        <h4>Rs 300</h4>
        <img src=" />
        <button>Add to Cart</button>
      </div>
    </div>
  );
}

export default App;
```

Using mapping:

```
import './App.css';
import {useState} from 'react';

function App() {
  const [cakes] = useState([
    {
      name: 'Chocolate Cake',
      cost: 500,
      image:  ‘'
    },
    {
      name: 'Butter Cake',
      cost: 300,
      image: ‘ '
    }
  ])
  return (
    <div className="App">
      <h1>Cakes</h1>
     { cakes.map((cake) =>(
      <div>
        <h3>{cake.name}</h3>
        <h4>Rs {cake.cost}</h4>
        <img src={cake.image} alt={cake.name}/>
        <button>Add to Cart</button>
      </div>
     ))
       }
    </div>
  );
}

export default App;
```

Styling UI

App.js

```
 return (
    <div className="App">
      <h1>Cakes</h1>
      <div className="cakes">
          { cakes.map((cake,index) =>(
          <div className="cake" key={index}>
            <h3>{cake.name}</h3>
            <h4>Rs {cake.cost}</h4>
            <img src={cake.image} alt={cake.name}/><br></br>
            <button>Add to Cart</button>
          </div>
        ))
        }
      </div>
    </div>
  );
}
```

App.css

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

.cakes {
  display: grid;
}

.cakes img {
  width: 10%;
}

.cakes button{
  margin-top: 0%;
}
```

Add to Cart Button functioning:

Create empty array to store items

```
  const [cart, setCart] = useState([]) //empty array
```

```
//pushing items to empty cart array
 const addToCart = (cake) => {
   console.log(cake);
   setCart([...cart,cake])
 }
  return (
    <div className="App">
      <h1>Cakes</h1>
      <div className="cakes">
          { cakes.map((cake,index) =>(
          <div className="cake" key={index}>
            <h3>{cake.name}</h3>
            <h4>Rs {cake.cost}</h4>
            <img src={cake.image} alt={cake.name}/><br></br>
            <button
            onClick={() => addToCart(cake)}
            >Add to Cart</button>
          </div>
        ))
        }
      </div>
    </div>
  );
```

```
return (
    <div className="App">
    <header>Go to Cart: {cart.length}</header>
      <h1>Cakes</h1>

      <div className="cakes">
          { cakes.map((cake,index) =>(
          <div className="cake" key={index}>
            <h3>{cake.name}</h3>
            <h4>Rs {cake.cost}</h4>
            <img src={cake.image} alt={cake.name}/><br></br>
            <button
            onClick={() => addToCart(cake)}
            >Add to Cart</button>
          </div>
        ))
        }
      </div>
    </div>
  );
```

Toggle to Go to Cart button:

```
  const [page,setPage] = useState("cakes");
```

```
 const renderCakes = (cakes) => (
    <>
    <h1>Cakes</h1>

    <div className="cakes">
        { cakes.map((cake,index) =>(
          <div className="cake" key={index}>
              <h3>{cake.name}</h3>
              <h4>Rs {cake.cost}</h4>
              <img src={cake.image} alt={cake.name}/><br></br>
              <button
              onClick={() => addToCart(cake)}
              >Add to Cart</button>
          </div>
        ))
      }
    </div>
    </>
 );
 const renderCart = () => (
  <>
  <h1>Cakes</h1>

  <div className="cakes">
      { cakes.map((cake,index) =>(
        <div className="cake" key={index}>
            <h3>{cake.name}</h3>
            <h4>Rs {cake.cost}</h4>
            <img src={cake.image} alt={cake.name}/><br></br>
        </div>
      ))
    }
  </div>
  </>
 );

  return (
    <div className="App">
    <header>Go to Cart: {cart.length}</header>
      {page == 'cakes' && renderCakes(cakes)}
      {page == 'cart' && renderCart(cakes)}
    </div>
  );
}

export default App;
```

OR ELSE:

```
const PAGE_PRODUCTS = 'cakes';
const PAGE_CART = 'cart';

function App() {
  const [cart, setCart] = useState([]) //empty array

  const [page,setPage] = useState(PAGE_PRODUCTS);
```

```
{page == PAGE_PRODUCTS && renderCakes(cakes)}
      {page == PAGE_CART && renderCart(cakes)}
```

NAVIGATION HOME AND CART:

```
<header>
      <button
       onClick={() => navigateTo(PAGE_CART)}
       >
       Go to Cart: {cart.length}
      </button>
      <button
       onClick={() => navigateTo(PAGE_PRODUCTS)}
       >
       Home
      </butto
```

```
const navigateTo = (nextPage) => {
   setPage(nextPage);
 }
```

REMOVE BUTTON:

```
 const renderCart = () => (
  <>
  <h1>Cakes</h1>

  <div className="cakes">
      { cart.map((cake,index) =>(
        <div className="cake" key={index}>
            <h3>{cake.name}</h3>
            <h4>Rs {cake.cost}</h4>
            <img src={cake.image} alt={cake.name}/><br></br>
            <button
              onClick={() => removeFromCart(cake)}
              >Remove</button>
        </div>
      ))
    }
  </div>
  </>
 );
```

```
 //pushing items to empty cart array
 const addToCart = (cake) => {
   console.log(cake);
   setCart([...cart,{...cake}]) // it doesn’t allow to duplicate objects
```

```
}

const removeFromCart = (cakeToRemove) => {
  //TODO: do sthg here
  //Using filter
  setCart(cart.filter((cake) => cake !== cakeToRemove ))
}
```

TOTAL PRICE:

```
const renderCart = () => {
   var total = 0;
   cart.forEach(item =>{
     total += item.cost;
   })
  return (<>
  <h1>Cakes</h1>

  <div className="cakes">
      { cart.map((cake,index) =>(
        <div className="cake" key={index}>
            <h3>{cake.name}</h3>
            <h4>Rs {cake.cost}</h4>
            <img src={cake.image} alt={cake.name}/><br></br>
            <button
              onClick={() => removeFromCart(cake)}
              >Remove</button>
        </div>
      ))
    }
    {total}
  </div>
  </>)
};
```

App.js

```
import './App.css';
import {useState} from 'react';

const PAGE_PRODUCTS = 'cakes';
const PAGE_CART = 'cart';

function App() {
  const [cart, setCart] = useState([]) //empty array

  const [page,setPage] = useState(PAGE_PRODUCTS);
  const [cakes] = useState([
    {
      name: 'Chocolate Cake',
      cost: 500,
      image: 'data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAoHCBUVFRgV
FRYYGBgZGBgaGBwcGBkaGhgYGhgaGRgYGhocIS4lHB4rHxgYJjgmKzAxNTU1HCQ7QDs0Py40NTEBDAwME
A8QHxISHzQrJSs0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NP
/AABEIAOEA4QMBIgACEQEDEQH/xAAbAAABBQEBAAAAAAAAAAAAAAAAADAAECBAUGB//EAEEQAAIBAgMFBQc
DAQUHBQAAAAECAAMRBCExBRJBUWFxgZGhsQYTIjLB0fBCUuFyFBVigpIWI6KywtLxBzNDU3P/xAAZAQAD
AQEBAAAAAAAAAAAAAAAAAQIDBAX/xAAhEQEBAAICAwEBAQEBAAAAAAAAAAAQIRAzESIUFRIhNhMv/aAAwDA
QACEQMRAD8A7zE6SeB0kcRpFgTMfrT40Fk3gxCPLTUZFpKRaI1HEajtl+hoJQxWsv4fQRTs70spGMdIjK
SiZVxWktGVcTpFRE9n/LLyyjs/SXljx6FIxRzGlAGtpB4HjCVtJDBcZP010RGIRGUk0HW0hJCrpFQjhtI
eCw+kKIToFGMeMYwrPrLIld/mEsCKBOKKKMObr6SOBk62kHgjmZl9X8aIhGgxCNKTUZFpIyLQNRxMvYbS
UsVLuG0inYvS0kYxLEZRGMq4nSWjK2K0ioh9naS+sz9naTQWPHoU8UeK0oAVdJDB8YWrpIYQayfoWhHiE
UojSFTSEkHioNR0hBIUtJMQnQKMY8YxgBvmlgSu3zSwIoEoooow52rpA4P5oepK2F+aZXtfxqLCHSCWEO
kpNMZFpKRaBqWKlvCHKVcVpLOE0inZ3pcSIxqcTSkmMr4nSWDK+I0ipw2zTlNFZm7OmksePRVKPFFKAdQ
ZSGGGsK4kMONZP0DiKISUokZF5KReFBqWkLB09JKKdAoxiJjMYwAT8UsCVc97QyyLxQ6lFGijJhVJUw/z
y28p0z8cyq501RCcIJYUaSk1GRaSmZtLayUgfhZyOCjjyJOQ1hbpUlt1BsVpD4PScbX9rHYNbDseW7drd
psBM2p7YYlV+VB0sQ3fnlI8pK1nDlfT04OFzJAHU2lHE7aw6fM4PRQWPdYTyxzicVnWqEJ+wG2X9P3nXY
REFNECBQqADO+nPrDz30r/ABk7bCe0aufgpOw0Jui27QWuPCDxHtAl9zdO9a5G8mXnOSxezArFkyvrb+J
V/uTEFC/uhqf0qrn/ABBdSOvGT5ZVU48Proj7WsjFURCbjWpnnpkoMuD2urWDLhg4PFal8xqCN24M85rI
QSDfqDeNRYqbqSD0y9Ipll+rvFj+O+r/APqBVQ/Hhd0dXYee5J0v/ULf0w9uu+GHkBMTYKtiS1FqlSm6r
vI6tqFPxKyN8LDO+gORzlPbWxsTSJIrYdlJsWKrTbvGhyleWWu2fhhvWnQ1fbysQd2jSvwu7fVR6w2zva
rGFrNhkN+RK5f1Ekd84AhxriaN8tCwA6XCZTraGyFVN/8AtO44ABLHep72XAkNY55369Ipcv07hhJ07pN
pVmUMuHz4g1Ey7CL38pcXHp+plU2uQx3SPG15w2ysQ7tutjqbAEXFMgXHK1w3fe0t7Wq4d0FNaytum53m
Ry9+jdeVj1mnle2N45vTsKeMpsbK6MeQdSfAGGZZ43XGGAuLh8/luF9Mu4mbWxfal0plFYPuXZAXV95Ba
6H9Vx00vobRTPfYy4bOq9KRLSVpibF9oExIbdyZCAynUX0PYec0q+Ppou87hR1OvQc+6aTKMrjZdVYtGM
ztobeoUd3fYgPoQrMAMs2Kg7ozlvD4xKihkdWU6EEEGG4PG9igSUYOI+8IySikd4R4BgPKQ+eXXlBj8Ym
VaRrJpCrpAocpkbd217v/AHVPOqwzPBAdCep4R26hTG5XUPjtrmx3BoSDkb5Zac5z2JxJsRbeOZCk3GfG
3AXvnK6neXdd2DW+cfAS4OZPAg59IDEUcQgsKha37gCfSY27deOEk9LlX2jJFim6QLDdNgByAtYCUEZHZ
dxN4s1tRkxNsyTlM+s9a/xbuvEan6mavsoHGIJc07EfCoO4GyOVmJ+K9m1/TF3VesY28V7PBEDb+84GYt
p03hrxnN7RxWJo/KqWH9Ry7bzfxu2wrFWIFsjZg3SZVXF0nObHwI/6Y7r4MfLXtW2d7Sp7xWcNbgAL2PM
```

```
gAFvzKb7e0tIOVKVd4HRU3vQ5d8xVoIw3VSw4kFbnwMzMVs6zEEseRJJB7CcovKw7jK7LaK4erT36lNlB
GTsoWx5Fr+UxqOwKTXdcQhQalRvm3SxzmRhHqUrhKjhTqN47p7VOXlI4vFNRIy3d4Xtu7oYNcNkLCxz+k
e9l43H6FtA+7qECowKH4WUFLjgwJN8wZn1MTvsAQHPN3OZ0JANrS4MfhKye7xAqpZgVdH96FyKkFGJbdz
vk17gcoJtjorf7rEJWB+VkYU2vaw30qMpBz4E/SVIm27Zb40jQBba/CQdbZzSw+JLjdvvndJyN2A0GRvf
MjTnM/aKOjMlQfFf4rqAwJ4XtoQQcjGwVdEFgpJ0LAZ+ZNvCVZJEy23tpe5qMM1CafE91XxIuOFxmeUh7
mwDbwBtnkV63UgE+I6yg4DZkEngXLm3YqWWEdN4DfLsOACWXuFxcyV+v1aw+zqjEblaiOjVSugNsmAOpz
ym5g8E9CzqwyzYmxBbdtdWGY48dDOWGFQfof/hHcM5bwWMqUj8CMUawdG+JHTiDy4Z8MjC3ZSael+zmJo
jef5HZRvkZ6X+UnTmRLuJwSVTvs3w2+djkBxIvlbynl77TZWb3aOmtlJzUdvGWsf7U13SkqmxRV3vhWxZ
QVBAtyzzvY6WhL+pyw97j03H4VHRSgJysupLA5gi3D80lfZmCXDb9RzuXGSX65nttcTA9mvbdW/3eJLob
EhwLq3+E2+IHsvN3FlK6q9GsrU7j3lySCjFVJsc13d4neOYKjOX67Z6s9XppLjusJ/eA5zPquKSJ70XZl
BJAIubC9wdDFSVKmhCiZ25J1i0P7wHONA/3an7/AEjw/ofyI8z6p+IS09QTLr1xvDOXUxuo4C3OgFz2AZ
zy2rtF6ld2sFDNcnib3Nr8gEtO72ntFFw7m/6G9LZTyHB4xmdjzqXHRQpS3mIsrttxTUb+Jrut3YFl/Vq
T/lAhsJtcZAOHHBWPxDpz7iJWR2bpGekOneAfWQ6GhVxaa7rAdoNj2m0zquJpsQN8rmpzByzBvzBHSVMT
s9X4Ect12XyzEpPswab9QDqVI9bw1C26baWLpFr76EknS+flKxdQbb6g8t6c+2EK5K7HoRkfG4jUt4GxR
WvyJU92dvKGocys+Ouw1Z9BdjwUgfXsvGfEu+a0xkOegvn2TAZgFFndCdVyOXdwiphyDuOexHYHS9rNa+
UWj3G8ivcX3VOWQzIBG8Dn04dY+0aZqKu+LkAAX1At6a6TlnquSVatUFuBBB45/N2wtDFoCAa1QC+er5d
AGFo9J3Gl/s/Tax07ud/4linsIU195c7mfxBiFJH6cjrpKVOqN6y1PeC3w7zBPitx3jcZ8rzapbLdxvNY
A2G6hJUddcybDPpBV12qYnDUK7h2Vt8IqXD2DhSbFjunOxt2ATndqbKbD1N0g7rAOh/cp07SLFTpmpnd4
bZNrXPXMaGXNsbAGJobgsHUl6TEWAJ+ZCeAYAd4BlY279sc5Nenlo7x4+d4RXbmfE+kavTZHZHUq6kqyn
UEGxEYm00ZCbzaXPZcnyjiqbEXFja+QvrwJgwZO/d3RAZKzboAtl8ugt13hrrGZzc3tfiNQe++cFr/AOY
j26c84aPYrvf9Ns7cdeYl7ZOOajUWopKlTbjmDkbjQrbIjr3zOXocz6SdIrytlz08uUVisb7eo7P21/bK
9BHVdx1rb4AJs6BAg3jppVtzAM6OrshF+S47zPM/ZCsq1qZY/LWB7vd1C2fHhl1nqrbRTgD3xSy9o5cdW
aVP7G0aWP7zTp4iKPU/Ue3EVMY54yjVqMTqYUtBOJy5bvb0sdTqJ4lC9JxxKN6G04bAC4BOpLA/5SLH/i
E9Bw4ynD7Rw606wpJmA7t46DusPCaYIy7X6DAHdF/GGdpQqISAR5a3EsYatv5HJhqPqJSbE2dQC7k7o6Z
k8hM6rVZ75bt9ARoB9c5OpjN9mT9KtYdwzJ77yG9BIdNTxOXfIqNxg41F7Z5eEKG46dsFUbLvjANaszaq
CO0yq7ZZADO+V4Zz6QNTSUKIiME3ixO9ewJJsOJtBmkp4Z+EsYzIIOSL6XMrBoQrEjQJ0Y9+ckj4hPkcp
/SzL6SIrWF/DmftC0nYqXOtwq9Li5MNjSymIqMbVMQ+Q03iZtJtegAA3vnsLZ1XHfqAPCc2QCbn68pNGt
mD0P25ESFxp7SWk+66b6lxZ98hjvIFF94AbwIse28q0aKfu4ab1jfjnB4ksVQnS72sMv0A2Gl8hAplofH
jNMZ6Y5XWXS2+GFrqADbO5GZudD2WjpgntvEHyt+feVd884QM9szcdovbl0hqlufiymEBBztbmL34ZD6n
POBehum1yfXvHCRRyNBpbyPL+JNsW+ZHp5Zw1RufgZoPy1z1I7CbwaF0bQ6EHs4G/GGbGuuoU9M841HF3
uAtjx5fmcXs5479Oi2PWpJhHLW96S7IbkEMB7qwA1yqg9/SZuJfE1Cd6o5Uk2G8QLXyGU0fZetvI1Bhkz
MQep3GF+X/ALRF/wDFOmpbMQaicvNyXG6kdXHhjlN5OA/ux+Z8THnof9hT9sUy/wBsvxr4cbPaAZIb3fW
Oqgay7kUkSpPYE20E5Gkgao7Nm28fA5/9U9B2Xs5quYFl/cePZznCbaIwmIem+e6+6D+5CN5T4Ms245lr
emGeWNupegxUKPbg2Y7RkR6ecgQjHk/C2ULi0DLvKdMx2/nkZSI3rOuTA5jqDLKVXoUvifSwYj/TlCm+9
a2Vss/y8hTrhXcH9xPcYY1V8eB/PKNKAsQCSbWvykKjqigsLu2YUn5RzP2ls2Iyscx17Jl4kguxPMgdLZ
DyAgEGrXzKjukRunn2STASdFFBt4+EYqW0Dcg9JRvNKqoO8O8fXylEhYQUImX9lMrMEfIM1gb2sSLKT03
t3xMpkL1k6Krcg3084yXcXhSjFWBBBIIPAwdMHTO06B3NWjTdyhYgqSVO8xU7uZBsTocxxmAXAJ4Znn9Z
Ctj4qiVSncEb2+QeZuoyHYB+GV1Q62PPu+s2MO4bC1VsWZGR0tmtt7ccHiDZ75/tHKZqYprWNiM7WFrcy
CJeO9Ms5PLYO/8AmcQt+fxLIrJbMZ5+GUIalMr8hvyJj3/xOv8Aql6yZbmb5fgltUpEZAehv2iM9OnpvE
dl7dfrDZ6v6pufz8EFvgH4eNvGW/cqdGHh166wVSgRmttPzSBdOi9j0viALXBBJ1+Er8StlpmLd89Cp0r
zzj2Pq1BiUCKWYqQRzXIt2WAJ7p6coMwzx3k6JnrFH3AihN084ovCF51yIBm7srYoyerpqF/7vtM/Y9Df
qfFoovbrwnTu+glcWEv9VHPy3H+cVlHFrDID0nkftPWXE1qzZFkY7vYDZbdMrd4nqFR8wJ5rtnYr0alSp
T+JCb2AJYXOYI5D7TbJjxalZGFqWAz+HQ9AdD3Xt2HpIY9GT41ORNm49h+nhKlByrlTYqeHAg6jzmm6by
lDqBkeY4E+hmbp7Y2LuGDjjke2RJmk2FTdK3NuRBuD2yi9LdO6fHmOEZdJ0KtivK4MrVkIcg8zDiieGcW
LT4x1URhVvJKZJkPKDvaA2KjkG/GRrU7ZjQ6faRDQtNxocwdeh5iAV5NBCVqNuo4GAtaMm3sLFWLUmsVq
```

```
D4QeD2srKeBOnXKZj07Ej81kEPEaiX8XTLWcZhhvX68fO8mnLuCbNySr/wDk3qJTvDbMqA1lRm3UcFGOW
jjd46azqn9kENtysw57yg9lrESsfTLls9ONLfn8xfmU2G2BUDspG6qXPvHBWnujU5XtCUfZmq6b9N6Trn
YqxGnCzLkZW0MXT6SdxNNvZnFAX9yT2MhPrnKFbBVKZs6OptxW3rABgcjlIsTlY9umX3jsCNQRnxGkY3G
Zva+oEDdP7B1lTGoWyO6VXqzWXdy53PlPVNpUN1rgZNn38Z5T7EknHUSoI+I3yJyKkG/KewbXPwrz3vK2
f0k63jTuWs5pjxRu6KZ6a7YGwawDkE/MMu0cPXwnQu2hnFohBuDYjMHiDOk2dtEP8LZP5N1HXpHx5T/yj
n47vyi64zvMrEpcMDxBmsdMhfmPt9pn1bXutyOI4r3TSxz43TzDaWEvVUrkEA3r5G4FtOHCWab3tbhp+f
mk7bHbLpVs2X4rfMpsfse+c3tD2cemC1I7620Pz36WyMiyunHkxrmmqm51OZ9Yxc8jC1Qf2kG+hH14cYV
adxe8FqRqEaZSdUllDcRe+mn8iEemYNk1HA69YBWFQxzIMhHZI2lFsnW0S3hEMl7oHTIwBI50OY5XhNxW
0yPXTxgSpXWTUwAb0yDYi01NkuzKaJOpul+fFel/p1lQNcWOcGtNgbqbWz74hPR8QjJVUMLEH/x6T1elS
3lB5gHxF5wWIVcVSOYWuguB+8Lmd3mbcNZ6NscFqNNmFiUW/hKxY8vQTYQMLMoIOoIuDCUMIEG6ihRyUA
DwE0lpSa0pWmG2ViadTcb3YXfsd3ePwg2yJ5iZGz9j4msjJjt1luCqqbG4N7kqfKdeKcmF5Q0cy0xMNsC
gjb601DZ3Y3Jz11JmlRwa6kdFW30ltE7/AEH3h91VG85t6nsEeoN2nwlEJc5Dix4AShjcZvtl8oyXs598
bFYsv8IG6g0HE9TABukzyy36jbDj17vZ7GKTuYpm1cUcuJgzVN8uGkgWjJ2zPTobOB22w+GoN4fuHzd/O
aSFHzRgx7bMO7WcoLSe/bMGx6TTHOsM+HG+56dM1M3zGfPQ9/Axivf5HwMxqG2aqDMhxyYZ+OsuUdu02y
dCvZZh4azSZSue8eURx+yaVXJ1seejfzMTEeyVs6b2PJh9RadVSxdFvlqKOhNvJpYFHlY9n8Q8ZRMssXm
WK2Niaf6C3VDceGXpM6ozr89Nx/lnrxofn5aQbCg6qD3Q8T/1eR0mVgSbi2o0I5ZeUE1E2DEWB/Lz1rEb
HpON10BHh5zKr+xeHY3DOvQPf1uY/GqnLPrzc0yI156InsNQ/fU/1r9oZfYbDf4z/nH2h40/9cXnSV+DD
eHmIUUFI3lJtexHI/gnoo9h8N/j/wBYk6XsNhh+89rkf8oh40f64vNhR6+cYhRq3nPWqPslhFtaglxxIJ
PidZpUNl0k+Wmi9iIPSLxpXmn48dwtEP8AKHLcN1WY36bs9Y9ng5oJvo6MotZxYkDQkTSLomrKveB6WlW
ptigujhjyUFj5XlTHxu7UZZ3KakXAIvz8tM07YB+Sm5/qso8Cb+Ug2Pqn9iDoCx8TYeULnjEziyvxrqhP
59pFsQi5FrnkvxW8Mh3zHJLfOzP2nL/SMvKSNQLyHlJvL+RpOD9rQfaLfoUL1OZ+w85WZmY3JJPMm8gjF
tAT3fUwiKeJA77+kzuVvbTHCY9HCRi0QQcyfL+ZMEcFHbr6wiqhvD8MeE32ih7S89dzaQD2kGjExab2pm
tEH6yGUY9IIuxg0ZiO+DkRYfggQgeEp1mX5WYdhI9JXMiXMommu1Kq6VG77H1nR0fe7it7z5lBzQWzHS0
4m5M39g7fTLD1SEdckJ+V14C/7hy4ysWXJNe5Gwa1YfqQ/wCUj/qMj/b6w4Ie9h9JYdenhnK7KOcpki21
qo/Qn+pv+2Q0263BE8W+0HVTqJXKQ3Tkx/Fr++a503B3E/WTTHYhv/kUdifcyoiS3Qpw9ndfi1Sou/zVn
/y7q/SaeE2JTb5t9/6qj+gIEBhbD+ZZxO3qNAfE924IvzHsUeplSSe6n+rdYhe0WzqFOmpWmgO8BoL6G+
Z1mCMQoyBHYPsJHae1TWYO6/0re4UdRoW6ysuJIyFgOlh6TDPKXLcdfHhZjqtFHY6K3f8AD/zZ+UcK3Eq
Oz4vtKIrdbwi1TI3V+LSpKvMntNvSFVlHygDuF/GZ6PLFJxDSbFo1CeccPILCLKlJJTCLAgyamUVFyikL
RQT6ecAmRJkiIF7cZLW1PeEffgGbl35emeUV4FsYvIFpCQLi/Z2yi0m1jrfxI9ImeVqmJA1IHaYI1idAT
1tbzbKA0tipAYtQ2ogS7cwO+58AAIN15sx8v584eikq1hdqYij8lUhf2v8AEvnmB2ETSp+2lb9dJH6oSP
IgjzmEthoB26nxMTPzMrypXjldJ/tep+ag47Ch+og29r6f/wBVTwT/ALpzbRboi8jnFHRH2vH6aL95Uel
5E+1lc/JTRf6mZvIW9ZgBYaksVyqpxYtR9q4mpk9VlHJBuDxHxecs4JANNeJ1J6k8Zl0TnNCi4EV99rkm
M9L71esdKv5/MovViWpIqpGktaGXEdPOZiVDeHRr85Oz01KVYy3TqzLpm0sUKh6DuEW03Fro5h0a+szkf
slmm/Xzl41Fi2DeEBErI0MD4S9M7Rt4cooHx8f4ij0W3nrHrAuR+ZwTB+ar2At5m3pBOg4l28T5KAPESW
vaVTEKMj6i/hI+/PBT3/D/AM1oJvh0AXwHpIBOJPXLP1ENxUxojO3EqOy5P0tBsy8Sx7TYf8NpFrWvn4f
eQ8YbHinvAZqAOwW85AsW58zx78o3bl9fCRIv+WhKdmiv1jFu2RdLcfD+ZG/fHstCXiOkgD0jiGykKSSL
dMkqRXJeOKQEKiGMg5QqkTPyaeIlJJbpJKyVLS0j34w8qVxgwongBCLR6+X1kfeESLvcW9JFtpzEcBRy8
oRKgvlKiJnn94cdPWHoe10m3hHU/n2gkXLh2Sag6i1uhilFi1TfnmPzlL9AzORTuk285dw2YBvy
jSpw6rK9AW5+H8ywD1msrHJLc/L/zFGv18hFKTp5vV4So+kUUj43natiPe6FheEQ+H1EQ+sUqdrFGkKFT1iikxV7WKHK3TJ
vSJ4RqOkUUZfQH1EQ+sUqdrDayaxRSiviiikxV7WKHK3TJ
Ede4xRTOKPS4wtLU/nKKKAWG0/1fWWMH+rs+piikzoVc/QeyGwfyiPFKx6ZZdtFvzyk107/pFFOnHpzZF
FFFKJ//2Q=='
    },
    {
      name: 'Butter Cake',
      cost: 300,
      image: 'data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAoHCBYWFRgV
```

```
FRUYGBgYGhoYGhgYGhoaGBoaGBoaGhgaGBgcIS4lHB4rIRgYJjgmKy8xNTU1GiQ7QDs0Py40NTEBDAwME
A8QHhISHjQrJCQ0NDQxNDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NP
/AABEIAKsBJwMBIgACEQEDEQH/xAAcAAACAgMBAQAAAAAAAAAAAAADBAIFAAEGBwj/xAA9EAACAQICBgY
IBAUFAQAAAAABAgADEQQhEjFBUWGRBQZxgaHwByIyQlKxwdETYqLhFHKCkvEjQ7LC4mP/xAAZAQADAQEB
AAAAAAAAAAAAAAAAQIDBAX/xAArEQACAgICAgEDAwQDAAAAAAAAAQIRAyESMQRBURMigTKhsRRhcZEFQ
lL/2gAMAwEAAhEDEQA/APH2d9rNzMgXO885Y1KF4D+EhSHQpeYOyO/wwm0oiAC9CkSbx5FmKloQRgZabt
MtJBYUBpYzTe0AohAY0A4uImfjCKaUzSjtiHhUmaUS0pIPHyYDgYTLiKBoSnSdvZVj2Aw5MKD3moxR6Gx
L+zRc9xjlPqnjD/st3wabDopnMA6idKepmMP+0YGp1Oxo/wBloqY7RylanEK1KdTierWKXXQfleVGJ6Oq
L7dNx2qZNMLKVWIMcpVrwVSgYDMRNAdf0N0u6ZX0huM6/BdIowzWxnmnR+JF7HXOu6MqXtLj8Byro7GlV
B1RgUdLWZWYJpZDEKozIHbNoxREpyYdKKjZCaUoMf1poU8tLSO5c5zWP66ypppoje2vlKtIjbo/q4hVF
2YCU2N600EtRuPO8Ti6tQ+u7GHhqHKRSnJc/QiykdfS2uVGpGHwj3f8ktcpMf3XxnxWAHbAm6oBGWmtpw
xRn/d6cW11B9xmM5NdYpYD2ec7fsfaWRl70+HqKRLKzTW1fWJbb+W9mEzvbWyxa5ZCoLgz9+Mp9p5HjS
oRebfuVEDHR6ZZNMWmMjZUGfjzyuDpxJd4H3T+ADb4X/p2KRLUO57QRPZfbT2gOSFrZhpWdhYQfk1MhjU
TCLULVuFxaRMY/aR2ULUYTyqp7Gxa+yQCjJZS0OGwXz/SUC8yTIKUQaPW/MZHrgnfmEjqkkMOvOD9Y0
IH6tzwc3EulLmPlqkaATfipIiW4BO96gN/zVYyMJh2Y1o5xEvr6BSGsaE/dHhFfOKOTYIX4n4uJmjTKMS
NIkZ6Wu9HlaOxcngGiNhnyzEU1cGbtpcZpiWCI4pPhHIaUhRCNeJCcWIzEfULmmfkhxHGm5U4LZG4+WCzDE
5DAF+3pMOKDtECV/tEF2Xx9nrTmJg3j7Aw6ELmJE75lxYNFmsYFsXafdIAcM0oXB1ObjVzt6Yi6QhtcWZ6
Gz9gaXl5Vz45vEBNFYg7V2KXqM4Ky8RDMyjv9+m4bk3q0NWWDnldn8d3f6izZXkZY0/Bx8d1ge/z7/4gXz9
SuRIoVXxXpOhW5hZQCq4JXK8knWP0vRgj2OSyrk+ViPG9eAYk1Zhq3hp/xWTp7NhuKCV5ahFM4OE1GmDA
d6hkZn6Vpbc5QUKG7zu37tHNPX4bCkcRRpmTnAsWtA/9f5Y/6FcVobw21xjU8XydRpRRe7AXO27KWZGn
uUH6/O5g9xRdfpE1a6ShU3l8jtaYdfOP5zY0IdMW+DgQW9GSfhCjDHfRvFZYdTURwSVCnO3/nSOCeFZs6sd4
0yoV6uS/l3P+I/5j+qVUQOg22UNiyjwDLCeo1q1oGtbzl2W6lYHVCj1KfKsB3p6eRKN5/CT/XY40R/Bwdrm
wT6wmnsW8ZnlIk6ss0KK8ieXE9zWmVK3TDAZUKEhaV14xDGq5XpP2IAkdhmC6m3YSGqEiPGOrGhM1/YSf
hyAjDJ8C2ONaQ5BZsWOtk2kdJOHjfBf3yMrwXcEY1pvVvHnyQPN2mDHxZtbv6LDJXf9aNmT7EWj7WbPLYN
lnR+Ioi0PoiEvWuwkufwq7yS9VepSnpDcelbS2ajaZDeXbx7OZLrWwn2EjrJR9xzSzJH2wukOC46i3FS/v
HCG/hi+bbjf9BRqRgCX7vXFqZagddCenk2vPwrZ7z+lyR+6m7TPGOIRLAeYpDsrMSf+24IhZtRM+w7+NuZ
u3fnsEnqhQ6q+bkIoIvzFoN89X7+R/zVfi+x7rOUPhSD9+LO9N2a3FDLdsbcj/J92XoDOWK+7i/lGHt8h
vgP9HdC1ZadE1g5OHfFoafyHb3LDyo96Z5Z1YnNvuA1z+ocr6DMhiRMZ+Jvxu3u8vlKR6WIfZB0h5cRc
12nN9dcFzpKdlz4lc0/lelX9nLL/ITlnCVVGyJZEe8oKxLlR/+ZyOeG4UZBrqHrrvctBp7JyvRwI/+yeh
6L0jhm4ShCo8rV4UNgFr60ONNp4I2hfnyJAtM+NBlbG7klEtBhjYEkNH3/PGq+9KTXtOiDZt6V1dzNu6dHH1R
uaq3oZBTjU+EjkYxLHUmHkE1FDN8VhZYhK1t4HHcY4dDOV1bHQLBSQVtG8FHZ/BFXGSyw7/pIGvkCeSx5Q3
+PfNCRBRYh6dMkVcpDmDgSsZqVg2Z3SMZBS5GOjhkV20YSYUNwbq+G03pm1dbddDvT9y3JEQW6e6k46rF+i
HjBtgSVZ/HpcpDdTphP7sZpdc4aWlvQzMdJMgPfm1oSgBIPCkCGjcZ8AaPNN/rXGbCmrGAijWSA3iRHxV1
AZ9dTTkNnAT3jhbeGC1G0G1N+Ah00tKvqxCbHOrz5R/5+a20/0N6FD9wWxLSyHxoSx2ElmUZ6TFMuTE8pvh
X/SinDJ9zNDhoNbdMyf/YzFytSxGrOdSdZ+7a9vRfhycD9FIb/dKuo/fADfz7OjRabXqpSzmyIHwmGi3a
6aKqV0lXi6UoDaLSlc/aqCSgvYaO6dnDIh5XG/Lkfkm3HHB4XiPAm4o+n96gMXqAEGAWJuKGUC1XbJcMnyNB
M6ybTjA3izxqN0aXx/n3dJfv5ZuaDIHxtrGp6WtRr4jQFDkK5hExaMsZGrkrrkUdaL6b0gqlB/FPAUL9U0b
Y8kUxVRpBBNkKakGcSDBtjVMHDEY/hUt1mqrNJUfXCIfdtrLevB54c0nSGgxaQp05cOh8JEUQoVJAqWnQnM
3mFVoK+49lyAIPDQDOnKhn19iJhsJ2oWfkiuA3o0DqjUuHRvjJy8WcH8v6ZAxo3GD6kmLY1xGNbdIySn8
q4wNaZvaDvtSb7VlhjJGTOpGOn7qw2O6Hqm6tEjTFHoCYeQiZhi9BbLZTgKJGPWWgzawvcS8fXKTdzobI
0MeoI2z/oA6Nu7LmtxR1ueUpnPwODAXqmvvcKVpBWRmyu4yEJZGP4aRy48n93IbmCkWFxxxUqzGSMS/jT
Oy4YV0ZNH0HTyNcBNTrCSSePF4Ooawg1wFNmJoqlg7GpqxDyJ8Gew5iXKrRadJ1VO7XrRXt13GChhM/7o8
h14tmJYzuTfhDR6mjXZB9YjDQkNZIZFsYGE9gEl2L58LSuvkwtawtjP2pEHqjV3OVGPZ7eIU1VOqiupQSfaVe
LGbEm5/H7pQlFfpWY+1m+OLB+fEz8EE9IMQSSCmhAQjR+K0X5OFWcWYIRSNZBKtLijc9sO8xqLqQDjjIuJHr
DHEHq0ekrDfRdNqTXyKSuBoYkPSZCAMFnv8KBl+xaEIOGJlvR8UjgkCJHfBHc7zyY4NW22P9VPpMWC+FC5H
CEBPnrh3TeMv2IcBdvLcGNo/NmtHy+XP1tB+uivbvsCCjE6NN3DUNzyX3nbKJ/jqWHxbYbzfl0v5hBxJEz
QzwGsJQsDcRk5lctrQp58iBYAu7F8q1CsWvKgWWoTyWGfYwv/y3JkJMcEo4GzNDzkfgMZJtg10LEnY4jWi
lLTXTHOnsfMXUHEYVfvUnfc/SJRPDYc8XmFNhe7gEw1e9oq3GU9wO/VRrAX+dlbiWYsvwqbCGnEvfWi0ZTgM
Ngz06ARqdXXBVLnlhFfiO6XTphCz8pz7g7GlFLunnB00Jj2A+vWRI+nNVyNoF/tIOwpnB3V2ibdVIqMF4Wlt
pD0V/Cvqpl7aKKmQRoSRTcYzDJ62tSs5rt9ljvn9vAZUtJbwP5fn0yGp2oTG3dG7X8AtWc8iqIiu1lyfH
Ctm75BdxG3kn8wlw/u+FcCMn21PB7sJSy3R8jn4MFgdfW18L8cy5VR3XUqZYhW+jBZ/ioV2+dB+IO62EGh
ajsLljDbV5ebw6J7fB/cFM5BDRF1Qxbvjoaxd2m2NVDZMO9XnRRzMSHhTNoAiwrdb2DDQbYqDV7Yn0LPu
GctDXaFt0IdCxiKFQ4qhIPyx4UL7Vb3TH8XHCzkCSAgDu2FsaRfwpKiMVTjfiwGLPuiaSdgVe0Ak8dNJtU
cO7oXXWxQg7Ae9cgGExoZAL1ulrZd2yGVrA7mRZJ06Aml6W8ZCbCiNG8pKpuiK5WPKk+0gS/KMPWOUmm8
o49rzQPDnqZj5yPIjmDZojGhuD8CxoIS2sqUEPyyXsdeNuJtjWiMt+tI+qyoBkYmxfpK1EYMd6r5sxzbk
Og8w+pzvAQHsQDUMJd9dBWjlRY2fChN0FmAJZAyBu7SvvDEAW3qOyYJL2lIrCzu5qI1T2/wNKjeTLrSQtJ6
pRC8uDv6B1eBcUeq6iUU4F3cMMgTXJtv+HVBPVumF42MS06nnEGXP8OHjNW9UqsQLrWs+Ayl5nCbh+3r6
yVJwp7IRhEhnd+Hr6MCotrpwvb9e6+kuyYXtQLa3E+u4JlOGSmmXCs5KIkZCbNGQFCQmXOwv/hTDXGNLHB
r99vxTHT4r9qnwIK6hw/yTUsrBdKEaGbOuHLEMg7v8XzSdwOnhsP1IKcu6hVOBlkjYY7eW4WW9GFsXOOiyuf
fgqxuaxbOlxBJY/zLhHjTMN99Id2oFu+y+3tmX7PJkLbSFpjAy5hpRZZJ0Xr7xsT+0cS8l0m+5eAqN3gXx
w0PVD6ck7WPw2ojSlFWIPEKLyPYkLHkJ9R4hq/QqTybh2cU4OOPuDA2uiZGYgOUyD+8mKyfyG7hCsrhkyL0e
2y7Gy0DSvkwRd5EgaCwAjOsGiIKWkJ7BUAepUBvB9yF+wGlNUq5iTFUtfSRHJq/XGh7gadvv2Rwxw6J9Sn
+pqFGw0wGS4AzzEAv9MgAwm8xQYzlSHNVfK/M9CLr9HCFR8QlrTSGqVDTV89wNI01vrdBhwiwVpsm9l7gIB
/pQbJCOuwj3o/Gf2Hfb8wLqdodWm+yv8MRvhpFT4/t0NtNpZ4Na+7eG8D8rL24kKIWm+xjsCPc4VNhffcu
iuyK7XTmGU3r1kmKYuG4ahgkdqDCuhhOxuBsiVAsNWJnSE8C6XjIzmn1/ZVBMUepRx2YRHQnUvxtH8x2RNS
QkOVjWbBSTiuQ8rrQUbsGJSaU7Pq4bsWMEeXM4D3qj9f+kNW3sGUK9YScfw3x5T9t+BmCU9vfsH7g0AoHQrm
rHn3zSfU6y3ryfHNm8hfkhCp7iGvn1smNBGcH01qDWnmBbHIjeC2hrHUoMpZnHLzrUemVyPyP/ahlGHk+8
dVJ2iDQDIHdNzBqA8iWRPj3kCGZ5gtwGz9d7TkuCv4ytkWnQjbA0c8ZUzQqXcODGMIkspnSF7gjNrnkBVDtq
yn8UrX8o+RObzh8y3bPxNXhvBjgEjybbfDsyNfMCs2wRK9Xf3XHmKAbuVpEvPFjOQrxlpPR5rzVOT5VFw7LSF
7DZjI8UmFqgtb6NJ9X97d7XrNP2wW3BkYgHSYMtS7O2zZgR+G2k/Y5NagV/GN6Xg6BK6d8dznQLy6rpLyhDG
1QwQFoXWb08V/hgO+c/lwV2rjD0qsW7zVq7C3VH0XXSD3Zdg/RKuU8ExqlUpnA8K8Y/V7t+/gpGcP/Izhtb
aB0EW8g8r8PbvmcMnLmiDkXQFMkl7HaY+uu0FoSnezoN9U9PpKYU8P3Ue9MgnLRjsZ5YDE8ry1oJy3dOEA
j8GiNdsV6QU0X+a5CJwcx0lV1f/lLLvsWLPVZQV0BbRCm80bO/OH5h5+h+7JRy0n7dOGX5DBCMQl1SKH6lAJ
3FRZBYdTmRXx2evR4xsIHgExDc1wL8B6KTRqlVLgp+KOa6qDdH3hjbE0eEqDAUbBFRmLjc+1J9CUTXNzF+c
+7dCTXrntlt6SNCz4bF8r8DAZQwCAbd6fHcG3IcjVjHc0a7H2Sls8+Q/cmQRxAm40bLjxyJHU0bCYO4a9k
42YRQz7mZaMvvrLQ7GGRufh8wpcXNqoAWHIc6zGRUDVQ/+9vSTtU76gUB3KvhHi61JcOSJhm3Ny7PxK4+Vb
9Zg7XZ8U9wcuVNbmxpXq/NbxOsPPpcfsCt7YGmSWCI2qjJ5Vl0BoeT85rQ7ZxlnB5Q4ERLXHvOs1rJePm
hn71T69Vu5c6ZNQ+0iRs2yiVNIGaDqCoTL5+x4K8dNPqSXq2GnRA09X9xNdZW1wGLpvHiZBUf3GAX8nGmO
bnbRMqbvghj7U8o6N92nu8Fw6W9qFHwwKHXMM3+KNi5mjH8pvdVDRq3G+0MFGAf5EFTaUjoEWI+B8OOPuU
m6+x2Y5RfpXgFhUKHmEO7Z1QJUGVRfgmTZVmgCy7FYE4JZowVRvQK41r7WESYp6+N7QYWv1KTZY2iQrw4CC
IzgYy4bhy27aQQq5FLjmECqeWxd3jDXUL9zwrdHCGuJrD2NgHyOHhYJCUqH85zQhjIvygKfeoUZf1eM0Rq
tnjx8vy2UUXhR7Ky9rXC9RLFUc1JWK8xRi+nRC2D6BVGxFdD9MA8k/hh4EzjwDHh6YJU1Ff2LqDPqOqGk5
xt9EOpMHx8A3k7zvqOe5CsiHh1c6BK2eGGiXhNx8Lk2C7vqyEhBIU1m1R/V2ZfvPeIwdx7D6SE2OgkvHTcv
qGV0o3v7T+kZ8hJUJF6eGcJFr+K59S9gdJjWkQJvRfywvdkg7n7GhK4KnMV/Kj/9vJ0ZB5wJPWBpmBchg
Ky7Qx3u5+vWtcaCNlEfNv8f5BEPFgPqd/qCf1O5dnRo1e5r/TD/VhBh9TlWFDnXr1ld1K3thZzC2jg9DE/ll
fA/AyRPWvUuqgb4JaA8sn6HETdqxmlbj3OJZ3QeRnERD1uuSVA9dVN0Z1hSypWF/x6Cy+e/sMhSMwMiN4/3
9obvxnrSDnj+FLy/ycwoOoZOfc5RHyjnRi3dWhTgJNb87+q+8PbdLMTpAMF4ue9W0PZyCtgz/OBQZSfjMzHf
jy9i9ubX5ECKQ5QmOjoeM8LkDOD0CP1Aa8uW9W7HU/vlv0RotGNJ1UZ1YiU0x+gMpmyWeX+uk39b4MvbUCm
nLnaZn9IHc4LZgKZ+0vSJ+2AMWsC1Nz8xqAT6qNYXWN+fuD3fDDFYHrBmhhBp2uYRc+D6GjLdXT8yrbiY0
sd6oE7Kz8MNQ0etw+BSzBrVMsq8uKvtqDtpntmbdaI5jLWOzkE5s0RLf+mPFh/mY4RdoGcG/yRmdE7iAKZNWK
```

```
R2iXCaYpRpFnRlhhxEaIlnRSdkI6OSbDkZCQYQ7rs83Hkc4I/uPPKerj1FI8yW2wFtnLzwPzivSGEWtTe
```

```
m49VxongfdPaDG389hyPgVMgw2HzfI+IB75YjwrG0Gpu9NxZkYqe0G0Er8J13pHwOjWSsBlVT1v509VvD
RnHKpiEFDTatIBTNgQAneZMAmRjKBmOqRem3GP8AqDbnIVK41BucgZU1S2+BvxjuIfjFGAk2ATAppVEX4
nUc2E+oCmiqqNQVRyH7T5k6KYCvSO6on/IT6gqC47voZa9gVOIXLzsWVeJTzz+0u66ef6ZX16f1+v3nPk
jZtCRSOIsX/wBRRvBHzlpiKPnv/eUuMbRdG865wZk0jpx02br0dFrjVCad4+aYYWgFwhv3zkkvZ248iem
H6OoF2CqM/OudDia64ddFSCx9o7ZHDUxh6ekbabeA3TnsdiixJMTl9ON+3+xDf1JUul+5DFYkuxkEGXnz
aBB2W7/pCl7TOK/7M3UaVIKm8xhW5xBHv2RmkZoJoIDvh1bOBVoRDGhMYkSJot4wiJeUQRUWEkHmmkaS5
37ZQBK2GDjcdh+h3ysegyNcZMORGvPeJf0KZMLi8DpLce0o5jaJGTC2uUe0SsnF0zOh634i6VrEGxXXYi
XSrYE7s/tKDoekwf1bAH2r7vvL7GHRS28gfX6Tpw5bxW/RyZ2uVL2EByB87ftBN8rjlpfYSa+yOz6H7yD
nM9p+v2nqYm3Ff4OKXbBuuztH/IfQQTG4vvHzW/zWGbX3/U/aAGodi/JpqI5P0h0Q2GD/AAVRydTfxAnm
YPET1Xrqt8DWvvon9azycqur7wV0L2FDSat2QKU92fObdR7ptHsNBQeImQTJua/ymQ2BWMg+E84CpTG4w
5qDfBO/5hIY0V9ZIAiO1CN/jFXkg0apsVIYawQR3Zz6l6NxAqUKVQZh6aN4D7z5Xnvnom6VFbAimT69Al
CPy618Dbulx7EzrHTwt4Gx8LRKrS89n+Bzlo6+fA+FjFnTz54WPdCUSoyKetS8+H2lH0zg7oSNa5922dW
9Ph58/SJ1aHnePP1nLmw8lRtCfF2UXR1XTQHbqPaJc4AgMpIuBKL+HNCoV9x8x27pb4Y5ieRuMqfo7HTV
oJ03XJPCc+2Z7v8AEuemDK3AVPX0DqN8rA55bdYyExl92SmbY3xhaFaWWubdby1qYME5SDYLdnNXBo0jm
ixCmsOhhxhiJgoRFckzSqdkNSF+6ZTVgbiMJSBN7RoTZFEMLDinYXPfJLRvqGW/hNEjJyFCl/mY3QwvCM
0cMMsvPkR+lSt7vPdNYQXsynlSBYehYQ5FlY7gTfsEZSmNsX6UfRW21jbu1ma5Hxg38HPzuVAuicPb1ry
eMfSqKo90Z9p/a3OEp1Vp0tI9w3k6gILo6kc3bWxJ56/tMcMHJLGve2RKW3J/gcOXdn55eMCfPfr+bcoR
z58+coJj54f4/wCU9mKo5WCdtZ4E+B+rwb5dwtyW3zMm539p7BmeZy7oJzv82zPiRKY0c714qBcI6n36l
NB/SC3/AFnmBA2Tt/SPjLGjQBzVWqMOLmy8gp5ziFqS0tEt7MKTWj2ZyWmJndeFBYB3sbXmQluyZFQWiq
J4eME4EZLH4TyMCwPwnlM2WJuBuizrHX7PCLuvDwksBSdd6N+sH8Ji10janVsj7hf2W7ibd5nKMOEhGiT
6zIuLjUcx9IFk88P2nCeizrcMRTGGrN/q0x6pJzdBkO0jUe4z0F1m6fJE9Cj0/Pnzqi7048ywbJIlEtMq
sVgw62I7OBldh6bIwVu4750bU4GphwdY/acfkeKp7XZvjzcdPooOl885XYBLOCdf7S96QwptqyG37yppE
o4Or955U8DhK5HbDIpRpFnJoOUihBhUHnVEpNGbRJU3+fJkxRW2Y12v9bSSLDWmqaZLbRBsOoGod+e6Ep
4ZRla/bnJol9cKiSk4v0Tzl8kPwbEAaoZEUbNndNqsMq7400uhOTZiIN0Ms0FhVWacm+jNmJKbp3EWdV3
DSPfkPlLivWCC7G3DaeAlQcG1Ri73UHUu22y+6RKEsqcF2ODSfJ9EKAaswZskXIDztlrpcoNECiwyG4TC
Z6Hj+OsUflvtmOSfJ/2NsfPnzbtgnbz558pjNBMeX2+g8Z0kGE/T/wAj6mCZ1ALMbIgLsTsVcye+bdu+/
PP6mcX6Qem9Bf4RCSxIasynVaxWn9T3cYkrGziOnsa2Jr1KzaQ02JAvqUZIvcABEkpcT3mHRWPx/p+syx
3t3gTQgD+HxM2Ey9o+H2hb8f0zLdnKIYIJxJmQpHGZCwEnffUHZowbG/v3/pH2kGxZJzCnjo3kP4tr+yn
9szKIu4+L9Ii7Px/SId6/5V5QD1vyiSFoWqHt5QDCNO/5YB24RAEwWLek61KbFXQ6SsNh+2yfQXUbrlTx
1MKxC10A003/AJl3ifOsYwGNei61KTFGU3DDzmOEtNok+qmSRKTiOo3pDp4oLSrkJX1blqcVJ28NfbO8K
zVNMXQuVkCsYKyJWDiOxcpEMX0Wj5j1TvGrlLYrNaMynijNVJFxm4u0cw+Eqp7ukN6Z811w2GxAOR5be8
ToNCQqYdW1qD2icE/+P9wf4ZuvIv8AUisoVRfXlGQlzfu/e0m/RSbNJewn63mL0bb2XYdtjMf6TMtUmV9
WL9k0WFprILhH+Mf2/vDJh3Hvj+395S8bJ8EOcfkmiQqIBILRO1z3WH0kvwF23P8AMT8tUuPiz+CXNG2q
qDa+e4ZnkJos51AKN7ZnuUfUiEFhkAAOAmi82j4v/p/6IeT4QJMOqnSN2b4mzPdsEkzTTNBs06oY4xVJU
Q5N9mHz589kgxmM0Ex8+dfyliMY+fOzjBM/n5ZfSSzY2GZnNdZOtC4cGnQs9c3BfIpT37fWbhq37pN30U
NdY+nBhUKoQcQ49UHMUwffbjuH0nmDo7MSxuxJJOdyTmSTfO81ULuxeoGZmOkxudIk673OuSRFB9lx57Z
SdCaM/h2y+hP1kVptZiQbKCdeeX+DGUVfz+Jk0pg3szDeNQy1axKFRX4ldAaThrXC3AvmdV8vOUkKN7Ze
J+0fpOHUEFuXhqkmBHvHw+0AEDhDw5zI8xI1sc89l/ETIUBx39I5mabsHMyZMjaYlkD/ACj+4/aDa3w+P
7Q5EiRAQuV/Kef7QLoNzeEbYd0GyxAKFRuPhI5cYwyQLJARFTbMXuNononU/wBJtXD2p4m9WmLANrde34
h49s86tNWjTEfU/RHTeHxKB6FRWB3HMcCNYPAx8rPlPA46pRcPSdkYbVNu47xwM9H6velqolkxSaa6tNM
m71OR7iOyaRn8io9jKzWjKboXrZhMUB+FWUt8Deq4/pOcvBLTTAHozYEnaatHQrI2mwZhmoqGSvN6Uhe
eFAT0pl5C8iWioAmlIlpAtIM8TAmzSDNIi51AmU/S3WPC4a4rV10vgT137wurvtFZRas+yJ9KdIUsOmni
Kgpg6l1u38qDMzzzpj0GikvxvZ6ncPZXXnE4nG1aj7s7E3Lbs7ZLCzsesPXypWpWKYWsWpB4YGjS2m4
/Ecfmb3RwHOcqKrn3/AJfaKqxOV/AfQQqE71Pd3QthobStUFvW+sMKrn3rdw+sVTVb1d+rPneFAOyw5/e
HJjpB/wANsrMRbd5zhrvsY8vvFLPvHbn95NVe+Wjbv8M4WwpDAR7qb+yQbG5DW2EboZ3YkHVa+S3Az4RM
```

```
U6hIzWwvlnnfVcwiU3G1f7vHVDkwpB6lRztA3HRvMiRWtc5rz/8AMyHJhSKIGbIhMO5uPqAdnGHxyAG4G
0fKSAnaR0fGHHu9og67m5z2W7t0BA2WQZZMzRgMCUg2SNPqHnfBGAhUrIFeEYaCgAMoZq0kZggBEZZzou
iuumNw9gmIcqPdf11/VmO4zn5GAHqPRvpgqrYV8Ojb2Rip/ta/znS4L0r4F/bFWmfzJpD9JM8JmRqTQqP
pHDdeOj3FxiqY/nOgf12ljS6awz+ziKTdlRPvPl2blc2FH1QMbTOqon94+802NpjXUTvcfefLAhCohyCj
6ar9NYZPbxNFe2on3lbieuuATXi6Z4JpOf0Az51EkIWB7djPShgU9ha1U8ECDmxB8Jz+P9K9U3/AwyJ+Z
y1RuQ0QPGeZiTWKwOg6T624zEXFTEPon3E/007NFLXHbKhRutBrCr9YAETuh1PZBIMz2fWHWAGKIdBNJC
mMCa2hAsAmyGGqIYZZNT5tACTMBh1bzlJC8CuqTWABReZI+flMjA//2Q=='
    }
  ])

 const addToCart = (cake) => {
   setCart([...cart,{...cake}])
 }

 const removeFromCart = (cakeToRemove) => {
   setCart(cart.filter((cake) => cake !== cakeToRemove ))
 }

 const renderCakes = (cakes) => (
    <>
    <h1>Cakes</h1>

    <div className="cakes">
        { cakes.map((cake,index) =>(
          <div className="cake" key={index}>
              <h3>{cake.name}</h3>
              <h4>Rs {cake.cost}</h4>
              <img src={cake.image} alt={cake.name}/><br></br>
              <button
              onClick={() => addToCart(cake)}
              >Add to Cart</button>
          </div>
        ))
      }
    </div>
    </>
 );

 const renderCart = () => {
   var total = 0;
   cart.forEach(item =>{
     total += item.cost;
   })
  return (<>
  <h1>Cakes</h1>
  <div className="cakes">
      { cart.map((cake,index) =>(
        <div className="cake" key={index}>
            <h3>{cake.name}</h3>
            <h4>Rs {cake.cost}</h4>
            <img src={cake.image} alt={cake.name}/><br></br>
            <button
              onClick={() => removeFromCart(cake)}
              >Remove</button>
        </div>
      ))
    }
    {total}
  </div>
  </>)
 };

 const navigateTo = (nextPage) => {
   setPage(nextPage);
 }
  return (
    <div className="App">
    <header>
      <button
       onClick={() => navigateTo(PAGE_CART)}
       >
       Go to Cart: {cart.length}
      </button>
      <button
       onClick={() => navigateTo(PAGE_PRODUCTS)}
       >
       Home
      </button>
    </header>
      {page == PAGE_PRODUCTS && renderCakes(cakes)}
      {page == PAGE_CART && renderCart(cakes)}
    </div>
  );
}

export default App;
```

FINALLY, PLEASE USE REFRACTOR TO MAKE YOUR COMPONENTS MORE MANAGEABLE.