

# Shopper Success Rails-React-Redux Code Exercise

## Assignment:

Implement an application **from scratch** similar to the one displayed below. This application performs the following:

1. Fetches a list of default desserts
2. Displays a list of desserts
3. Adds dessert items; data persistence (e.g. a database) isn't required

4.

You can zip up the code files when complete and submit them to [dj.burdick@instacart.com](mailto:dj.burdick@instacart.com)

## Requirements:

- UI built with React and Redux
- Must use webpack
- Styling (layout and typography) should be similar to Design (see below image)

## Optional (but recommended):

- JS tests (any library will work, although Jest is preferred)
- Using Typescript

## Additional details and tips:

Design Requirements:

- 1) Make a grid layout for all the desserts
- 2) Should be responsive for mobile - desktop widths
- 3) "Read More" - should be collapsible content

Desktop View & Tab / mobile view:

1st Dessert should be that large image. Followed by other desserts in a grid system below it. Everything changes into list layout on tab / mobile widths



4.00

## Cheery Cherry

[Read More](#)



3.50

## Pancake Plate

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ...* [Read More](#)



4.00

## Sweetie Cakes

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ...* [Read More](#)



5.00

## Pumpkin Burrito

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ...* [Read More](#)



### Cheery Cherry

4.00

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ...*

[Read More](#)



### Pancake Plate

3.50

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ...*

[Read More](#)



### Pumpkin Burrito

5.00

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ...*

[Read More](#)



### Sweetie Cakes

4.00

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ...*

[Read More](#)



### Cheery Cherry

4.00

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ...*

[Read More](#)



### Pancake Plate

3.50

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ...*

[Read More](#)

1. You can use The following json as your local db to start with
2.
  - a. For adding a dessert, pick any dessert that you like and an image url (use google). You can use the same sample description

- b. Design a modal to add a desert with form and required fields (name, price, imageUrl, description) - make it look pretty

```
[
  {
    "name": "tiramisu",
    "price": 5,
    "imageUrl":
    "https://imagesvc.meredithcorp.io/v3/mm/image?url=https%3A%2F%2Fimages.media-allrecipes.com%2Fuserphotos%2F9354028.jpg&w=426&h=285&c=sc&poi=face&q=85",
    "description": "Lorem ipsum dolor sit amet, consectetur
    adipiscing elit. Praesent ut lacus rhoncus, vulputate urna aliquet,
    semper ex. Praesent id urna feugiat, tempus purus sed, blandit ante.
    Vestibulum a posuere ipsum, vel vestibulum nibh. Etiam id purus in
    purus scelerisque iaculis. Praesent in leo at felis porttitor
    placerat. Aenean lectus risus, aliquet at semper in, malesuada eget
    velit. Duis vitae lectus scelerisque, condimentum ligula vel, cursus
    lectus."
  },
  {
    "name": "Red velvet cake",
    "price": 6,
    "imageUrl":
    "https://imagesvc.meredithcorp.io/v3/mm/image?url=https%3A%2F%2Fstatic.onecms.io%2Fwp-content%2Fuploads%2Fsites%2F43%2F2021%2F08%2F19%2FSouthern-Red-Velvet-Cake161-2000_crop.jpg&w=426&h=285&c=sc&poi=face&q=85",
    "description": "Lorem ipsum dolor sit amet, consectetur
    adipiscing elit. Praesent ut lacus rhoncus, vulputate urna aliquet,
    semper ex."
  },
  {
    "name": "Chocolate vanilla pie",
    "price": 7.5,
    "imageUrl":
    "https://imagesvc.meredithcorp.io/v3/mm/image?url=https%3A%2F%2Fstatic.onecms.io%2Fwp-content%2Fuploads%2Fsites%2F43%2F2018%2F04%2F13%2F729302_cropped.jpg&w=426&h=285&c=sc&poi=face&q=85",
```

```

    "description": "Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Praesent ut lacus rhoncus, vulputate urna aliquet,
semper ex."
  },
  {
    "name": "Peach cobbler coffee cake",
    "price": 4.5,
    "imageUrl":
"https://imagesvc.meredithcorp.io/v3/mm/image?url=https%3A%2F%2Fimag
es.media-allrecipes.com%2Fuserphotos%2F9354028.jpg&w=426&h=285&c=sc&
poi=face&q=85",
    "description": "Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Praesent ut lacus rhoncus, vulputate urna aliquet,
semper ex."
  },
  {
    "name": "Chocolate chip cookies",
    "price": 2.99,
    "imageUrl":
"https://imagesvc.meredithcorp.io/v3/mm/image?url=https%3A%2F%2Fstat
ic.onecms.io%2Fwp-content%2Fuploads%2Fsites%2F43%2F2019%2F12%2Fbest-
chocolate-chip-cookies.jpg&w=300&h=201&c=sc&poi=face&q=85",
    "description": "Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Praesent ut lacus rhoncus, vulputate urna aliquet,
semper ex."
  }
]

```

3. Fetching a list of desserts must display a "Loading..." screen (can be a spinner). Initially screen should just say fetch desserts and then on button click - it should fetch all desserts from db. You can simulate the loading by using promises. Display all desserts and hide the button
4. You can use any CSS you want, so long as the result matches the image. (Flexbox is encouraged)

5. For the purposes of this exercise, redux should be treated as the single “source of truth” for managing **all** application state.. Craft each relevant UI piece as a **pure functional component**.
6. It’s not required, but using a library such as ImmutableJS or immer is often helpful to create immutable objects within the redux store should use of “. . .” spreads in the reducers become hard to manage.
7. Handle all other edge cases and error states that might happen

8.