

# React Mock Code Challenge: Plantsy

## Demo


Use this gif as an example of how the app should work.

# Plantsy

### New Plant


Add Plant

**Search Plants:**




**Aloe**  
Price: 15.99

In Stock




**ZZ Plant**  
Price: 25.98

In Stock




**Pilea peperomioides**  
Price: 5.99

In Stock




**Pothos**  
Price: 12.11


In Stock



**Jade**



**Monstera Deliciosa**



**Fiddle Leaf Fig**

## Instructions

Welcome to Plantsy! You've been tasked with building out some features for the admin side of a plant store. The designers have put together the components and CSS. Now it's up to you to bring the features to life by adding stateful logic as well as persisting data to the backend via our API.

Your job will be to make our app work according to the user stories you will find the [Core Deliverables](#) section.

## Setup

1. Run `npm install` in your terminal.
2. Run `npm run server` . This will run your backend on port `6001` .
3. In a new terminal, run `npm start` .

Make sure to open <http://localhost:6001/plants> in the browser to verify that your backend is working before you proceed!

## Endpoints

The base URL for your backend is: `http://localhost:6001`

## Core Deliverables

As a user:

1. When the app starts, I can see all plants.
2. I can add a new plant to the page by submitting the form.
3. I can mark a plant as "sold out".
4. I can search for plants by their name and see a filtered list of plants.

## Endpoints for Core Deliverables

### GET /plants

Example Response:

```
[
  {
    "id": 1,
    "name": "Aloe",
    "image": "./images/aloe.jpg",
    "price": 15.99
  },
  {
    "id": 2,
    "name": "ZZ Plant",
    "image": "./images/zz-plant.jpg",
    "price": 25.98
  }
]
```

## POST /plants

Required Headers:

```
{
  "Content-Type": "application/json"
}
```

Request Object:

```
{
  "name": "string",
  "image": "string",
  "price": number
}
```

Example Response:

```
{
  "id": 1,
  "name": "Aloe",
  "image": "./images/aloe.jpg",
  "price": 15.99
}
```

## Advanced Deliverables

These deliverables are not required to pass the code challenge, but if you have the extra time, or even after the code challenge, they are a great way to

stretch your skills.

You'll have to add additional elements for these features. Feel free to style them however you see fit!

Note: If you are going to attempt these advanced deliverables, please be sure to have a working commit with all the Core Deliverables first!

As a user:

1. I can update the price of a plant and still see the updated price after refreshing the page.
2. I can delete a plant and it is still gone when I refresh the page.

## Endpoints for Advanced Deliverables

### PATCH /plants/:id

Required Headers:

```
{
  "Content-Type": "application/json"
}
```

Request Object:

```
{
  "price": number
}
```

Example Response:

```
{
  "id": 1,
  "name": "Aloe",
  "image": "./images/aloe.jpg",
  "price": 16.99
}
```

### DELETE /plants/:id

Example Response:

{ }