```
//title: posters+

// description: a poster purchase store where someone can purchase a poster
// for there home or computer desktop

// firebase data base example
 // {
   // name: imageName,
   // imgURL: URL(URL to an imgBB hosted image),
   // description: description for image,
   // cost: 10.00
 // }

// Image Hosting: https://imgbb.com/ I have tested it and you can upload your own images and the
website gives a specific url to that image.

//MVP: Display 3 posters with a title image and description and have an add to cart button then have a
button that takes them to there cart page that shows what they picked and totals and stuff.

// Stretch goals:
   // 1.  15 Different posters with a title, image, and description
   // 2.  Create a fake checkout process
   // 3.  Create a fake admin page where the administrator can update/create new poster data without
going to firebase

// what componenets I will use
// what will go in my states

// pseudo code:

   // create states and use effects to hold data from firebase and properly organize it

   // first state will be used to hold an array of the firebase data poster objects so something like this:
   // [{name:SomeName, imgUrl: someUrl, description: someDescription, cost: someCost },
{name:SomeName, imgUrl: someUrl, description: someDescription, cost: someCost }, .... ]
```

```
    // second state will be used for the cart objects witch will also be an array and only use a specific section of the fireBase database a section with a key of cart essentially, for objects that are in the cart it will look the same as above:
    // [{name:SomeName, imgUrl: someUrl, description: someDescription, cost: someCost }, {name:SomeName, imgUrl: someUrl, description: someDescription, cost: someCost }, .... ]

    // I will have a useEffect that will run on load that gets the fireBase database and every time on change update the proper usestate with the new database

    // For the MVP I will use minimum 2 components one to pass the first State and have it return a div with the posters to be displayed and another component which will use the second state to create the cart information and return it as a div

    // use a method/function to loop through the data and put data onto pages with other required information

    // function for when button clicked that stores selected poster data into its own object that is put in its own firebase cart object. I will then have button event handlers to add objects to cart. For example, if user clicks add poster 1 to cart then what would happen is the button would run a onClick function which would push the {name:SomeName, imgUrl: someUrl, description: someDescription, cost: someCost } onto the specific cart database section in firebase.

    // when show cart is clicked new component will be added as either an overlay or by removing all the posters on page then looping through the firebase cart array and displaying all the cart information necessary with an exit button to exit cart
```