

Looping through an array in JSX to display NewsItems from state | Complete React Course in Hindi #26

In the previous videos, we have seen how to set states and props in our News Monkey application. In today's video, we will populate the content and desired images in our application. So without further ado, let's begin:

Rendering Articles in NewsItems

Earlier, we have passed the articles of sample response in the components of News.js. We have set the articles array as a variable and have used it as a state. Now, We want to iterate the articles array present inside the state, and then we would render these articles in our NewsItem Component.

Iterating the Array

At this point, the title of all `NewsItem` Components is 'mytitle'. We have to change it with the title of the News Element. In a similar way, we want to change the description and Image of the `NewsItems`.

```
render() {  
  return (  
    <div className="container my-3">  
      <h2>NewsMonkey - Top Headlines</h2>  
      <div className="row">  
        <div className="col-md-4">  
          <NewsItem title="myTitle" description="mydesc" imageUrl="https://cdn.24.co.za/files/Cms/General/d/10743/97d776dc91734e98906c0e1b7f3b1afa.jpg" newsUrl="T000"/>  
        </div>  
      </div>  
    </div>  
  )  
}
```

Description is set as 'mydesc'

Title is set as 'myTitle'

Image URL

Figure 1.1: Passed Values in Title, Description, and Image URL

The title, description, and URL to an image of an article are already provided in the sample response as shown below:



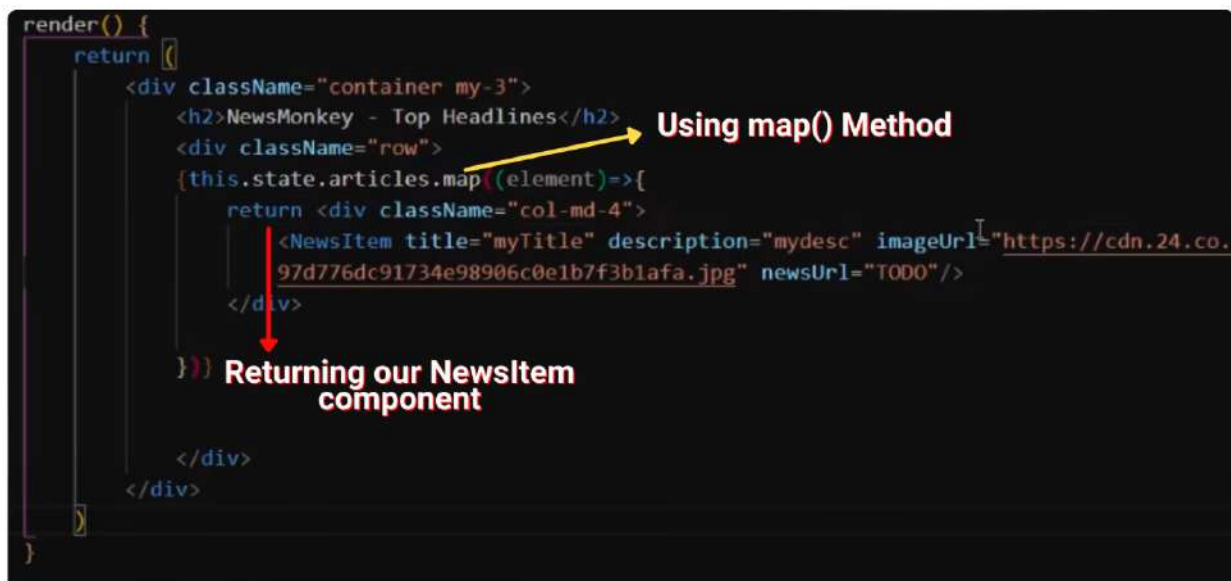
Figure 1.2: Provided Title, Description, and URL to an image

We want to change the values of Title, Description, and Image URL from the previously passed values (Figure 1.1) to the values provided in the sample response of the article.

Getting Started

To do that, first of all, we have to iterate the articles array and then render its data in the `NewsItem`. We can do so by using the `map()` method.

`map()` method: The `map()` method is the most commonly used function to iterate over an array of data in JSX. You can attach the `map()` method to the array and use a callback function that gets called for each iteration.



```
render() {  
  return (  
    <div className="container my-3">  
      <h2>NewsMonkey - Top Headlines</h2>  
      <div className="row">  
        {this.state.articles.map((element) => {  
          return <div className="col-md-4">  
            <NewsItem title="myTitle" description="mydesc" imageUrl="https://cdn.24.co.97d776dc91734e98906c0e1b7f3b1afa.jpg" newsUrl="TODO" />  
          </div>  
        })}  
      </div>  
    </div>  
  )  
}
```

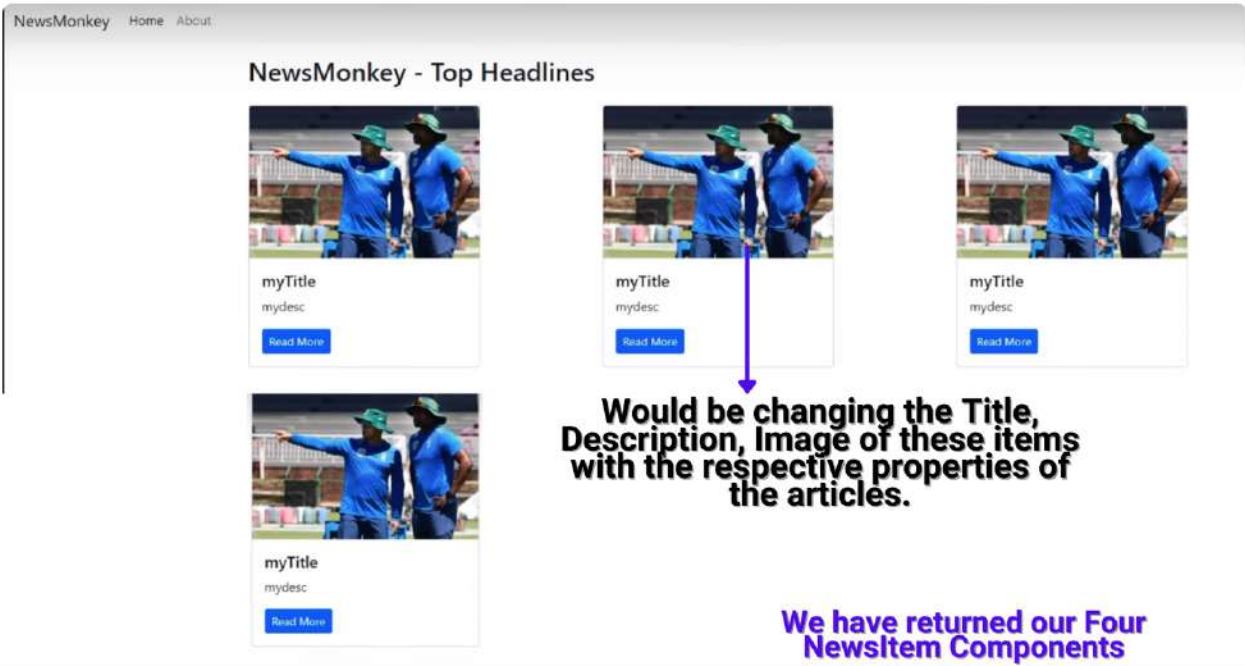
Using `map()` Method

Returning our `NewsItem` component

Figure 1.3: Returning the `NewsItem` Component

Explanation: In our case, The map would go through each of the objects in the array, and it will make each individual array item available to us as an element. Notice, Inside the arrow function, we have returned the NewsItem Component.

Result: The four News Items Components get added to the News Monkey Application. We haven't passed the values yet, so all of the components are having the same Image, Title, and description as shown below:



Passing Title, Description, and URL to Image

Now, let's begin passing the provided values of articles in our NewsItem Components. We would be passing the title, description, and URL to the image in the respective props of the NewsItem Components.

```
render() {  
  return (  
    <div className="container my-3">  
      <h2>NewsMonkey - Top Headlines</h2>  
      <div className="row">  
        {this.state.articles.map((element) => {  
          return (  
            <div className="col-md-4" key={element.url}>  
              <NewsItem title={element.title} description={element.description} imageUrl={element.urlToImage}>  
            </div>  
          )  
        })}  
      </div>  
    </div>  
  )  
}
```

Title as Article Title

Description as Article Description

Image URL as Articles Images URL

Figure 1.5: Passing Title, Description, and URL to Image

Result: Hence, we have passed the provided values of title, description, and image to our four NewsItem Components. Here's how our NewsMonkey application looks like:

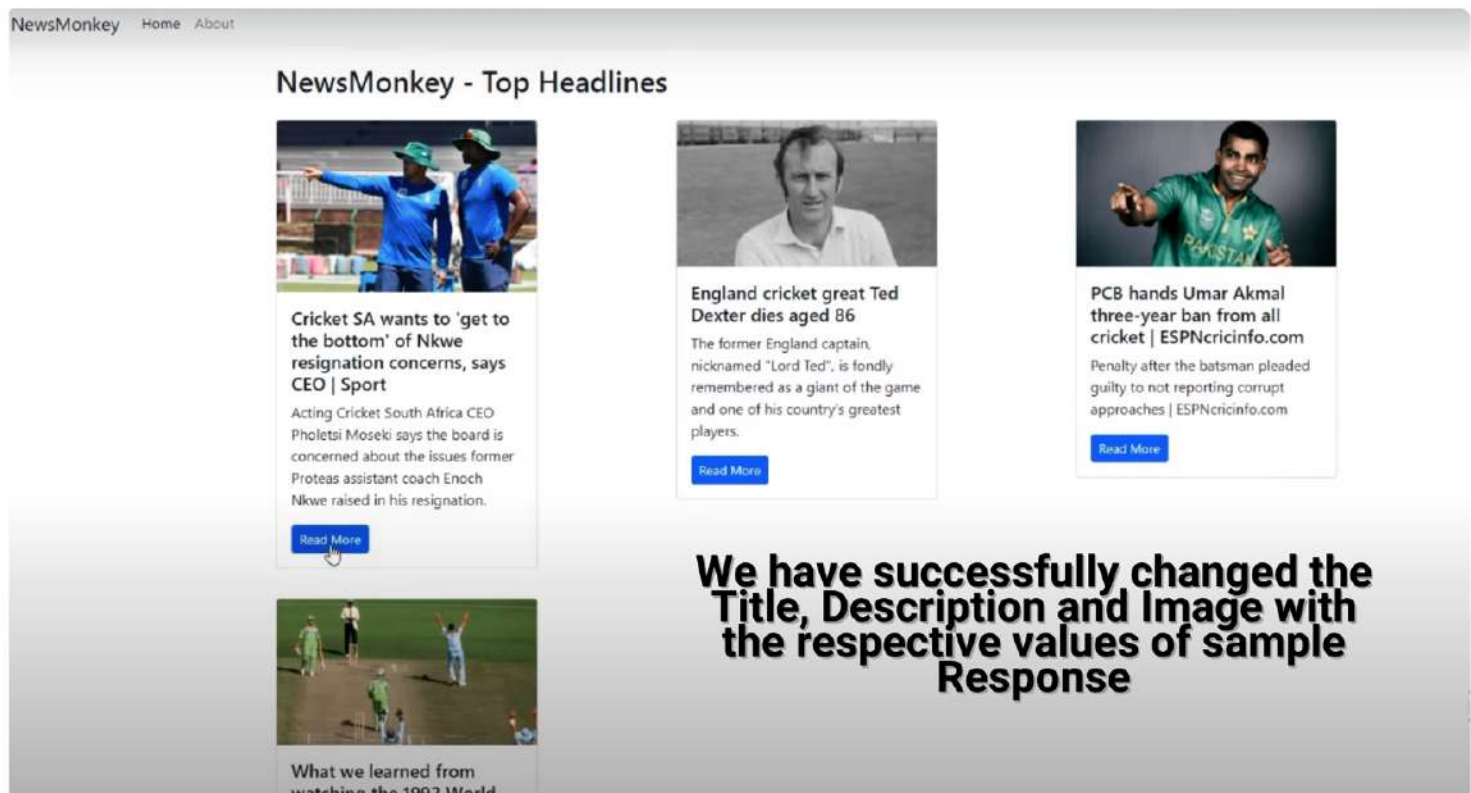





Figure 1.6: Our application after passing the values

Making the 'ReadMore' button Functional

We would like to display the complete News in a New Tab whenever the user clicks on the 'ReadMore' button. To do so, we would add a `newsUrl`. We would be passing the value of the URL of the respective article to `newsUrl`.

```
components >  NewsItem.js >  NewsItem >  render
import React, { Component } from 'react'

export class NewsItem extends Component {
  render() {
    let {title, description, imageUrl, newsUrl} = this.props;
    return (
      <div className="my-3">
        <div className="card" style={{width: "18rem"}}>
          <img src={imageUrl} className="card-img-top" alt="..." />
          <div className="card-body">
            <h5 className="card-title">{title}</h5>
            <p className="card-text">{description}</p>
            <a href={newsUrl} target="_blank" className="btn btn-sm btn-primary">Read More</a>
          </div>
        </div>
      </div>
    )
  }
}
```

Using NewsURL as Props

Open Link in New Tab

Figure 1.7: Using NewsUrl as a Prop

Above, we have used `NewsUrl` as a prop and have even assigned the link of 'Read More', which is the functionality to open it in a New Tab. Here, the '`a href`' is equal to the news Url. We would be passing the URL of the respective article in it from `News.js` as shown below:

```
render() {  
  return (  
    <div className="container my-3">  
      <h2>NewsMonkey - Top Headlines</h2>  
      <div className="row">  
        {this.state.articles.map((element) => {  
          return <div className="col-md-4" key={element.url}>  
            <NewsItem title={element.title.slice(0, 45)} description={element.description.slice(0, 88)}  
              newsUrl={element.url}/>  
          </div>  
        })}  
      </div>  
    </div>  
  )  
}
```

Using Slice Method to limit the length of characters in the Title and Description

Figure 1.8: Passing the Url in the prop

Enhancing the Structure

The length of all the NewsItems Cards isn't equal due to this, our NewsMonkey application is looking destructured. Therefore, To fix this issue, we would limit the length of visible characters in the title and description of the NewsItems Cards. This can be accomplished by using the slice method of Javascript as shown below:

```
<h2>NewsMonkey - Top Headlines</h2>
<div className="row">
  {this.state.articles.map((element)=>{
    return <div className="col-md-4" key={element.url}>
      <NewsItem title={element.title} description={element.description} imageUrl={element.urlToImage} newsUrl={element.url}/>
    
```

Passing the respective article
URL to newsurl

Figure 1.9: Using slice() Method

Our Application

Here’s how our application looks like after so much enhancements:

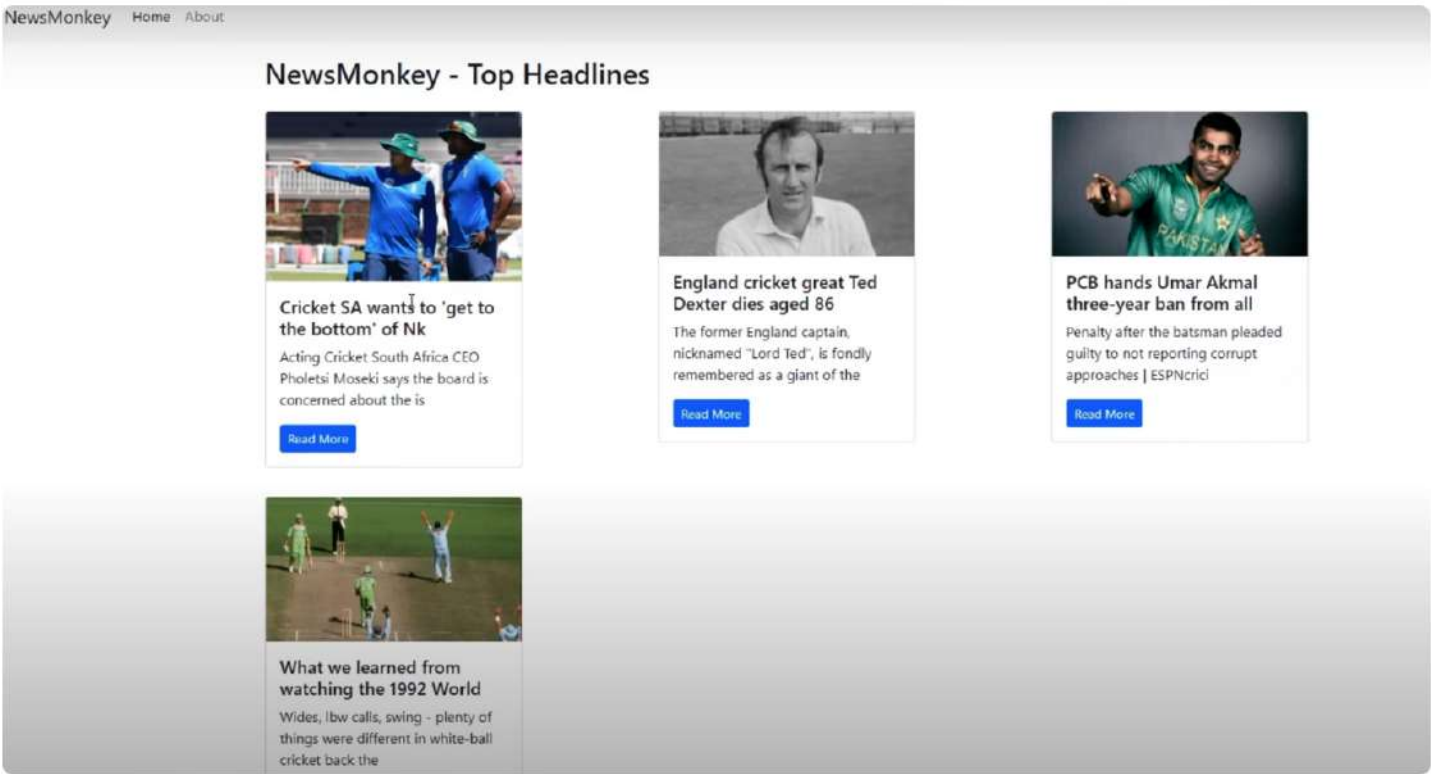


Figure 1.10: News Monkey Application