Using Fetch API in React to populate NewsItems | Complete React Course in Hindi #27

In today's video, we will be rendering the news from NewsAPI in the NewsMonkey application. So let's Populate our NewsItems using the NewsAPI:

News API

First of all, you have to visit the News API website. You will get a lot of the latest news categories, which you can use in your application. We would like to render the 'Top headlines in India' in our NewsItems. For that, we will copy the URL of 'top business headlines in the USA' and will use the filter to remove the business category and select the country as India.

Removing the article variable and State

Till now, we have created an article variable and have used it in our state. Now, we would remove those articles and the created state, as we don't require them anymore. As of now, we will render the data in our application from the endpoints. For that purpose, we will change the articles with the help of the componentDidMound method.

componentDidMount() method

This method gets invoked once the component has been rendered.
As a result, the constructor of our application gets executed first, followed by the 'render method,' and at last, the ComponentDidMount() method is invoked.
Note: ComponentDidMount() is a lifecycle method. We will be discussing the lifecycle methods in detail in the upcoming videos.

Fetch data in React using the async/await syntax

First of all, we would use our copied News URL in the component Did Mount method as a variable. This URL will help in fetching the News in our application.
Code:

```
componentDidMount(){
    console.log("cdm");
    let url = "https://newsapi.org/v2/top-headlines?country=in&apikey=dbe57b028aeb41e
}
```

After that, we would be using the Fetch API. Remember, the fetch API takes a URL and returns a promise. Here, how you can use the fetch API:-

Code:

```
componentDidMount(){
    console.log("cdm");
    let url = "https://newsapi.org/v2/top-headlines?country=in&apikey=dbe57b028aeb41e
    let data = fetch(url);
}
```

Using Async and Await

Async keyword is used to make a function asynchronous. Async can wait inside its body to resolve for some of the promises. The await keyword will stop the execution until a defined task is completed. In our case, it will wait for the promise to be resolved.

Code:

```
async componentDidMount(){
    console.log("cdm");
    let url = "https://newsapi.org/v2/top-headlines?country=in&apikey=dbe57b028aeb41e
    let data = await fetch(url);
    console.log(parsedData);
}
```

Once, The promise is resolved, it would provide us the data. We can parse the data as text or JSON. After that, we would use the setState method to set the state of the articles as parsed Data articles.

Code:

```
async componentDidMount(){
    console.log("cdm");
    let url = "https://newsapi.org/v2/top-headlines?country=in&apikey=dbe57b028aeb41e
    let data = await fetch(url);
    let parsedData = await data.json()
    console.log(parsedData);
    this.setState({ articles: parsedData.articles })
}
```

Resolving “Cannot Read property ‘slice’ of null” Error

This Error occurs as some of the values, such as title, description, are null for some of the articles. Due to this, we can’t use the slice() method to set the limit of the characters. To resolve this issue, we would be using the ternary operators as follow:

```
<div className="row">
    {this.state.articles.map((element)=>{
    return <div className="col-md-4" key={element.url}>
        <NewsItem title={element.title?element.title.slice(0, 45):""}
            description={element.description?element.description.slice(0, 88):""} ima
            newsUrl={element.url} />
    </div>
    })}
</div>
```

Explanation: Now, If in case the element title or description is Null, then a blank space will be rendered, that is the else statement. Otherwise, the sliced content will be rendered in our NewsItems.

Our Application: All the latest articles get populated in our News Monkey Application.

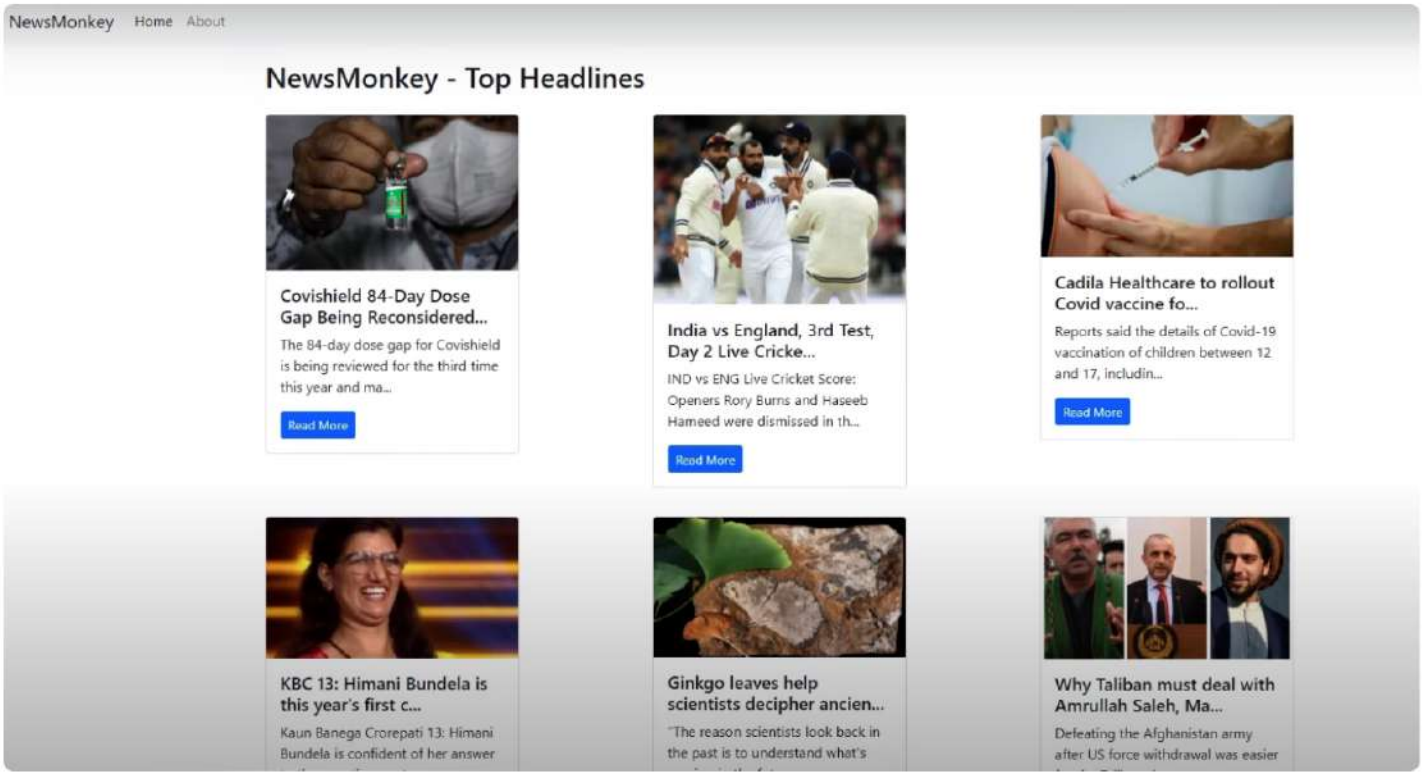

Figure 1.1: NewsMonkey Application

Resolving the 'Image Not Rendering' Issue

Our application is quite astounding, but for some of the NewsItem Components, the image isn't rendering in the NewsItem. Let's fix this issue and wrap up this tutorial:

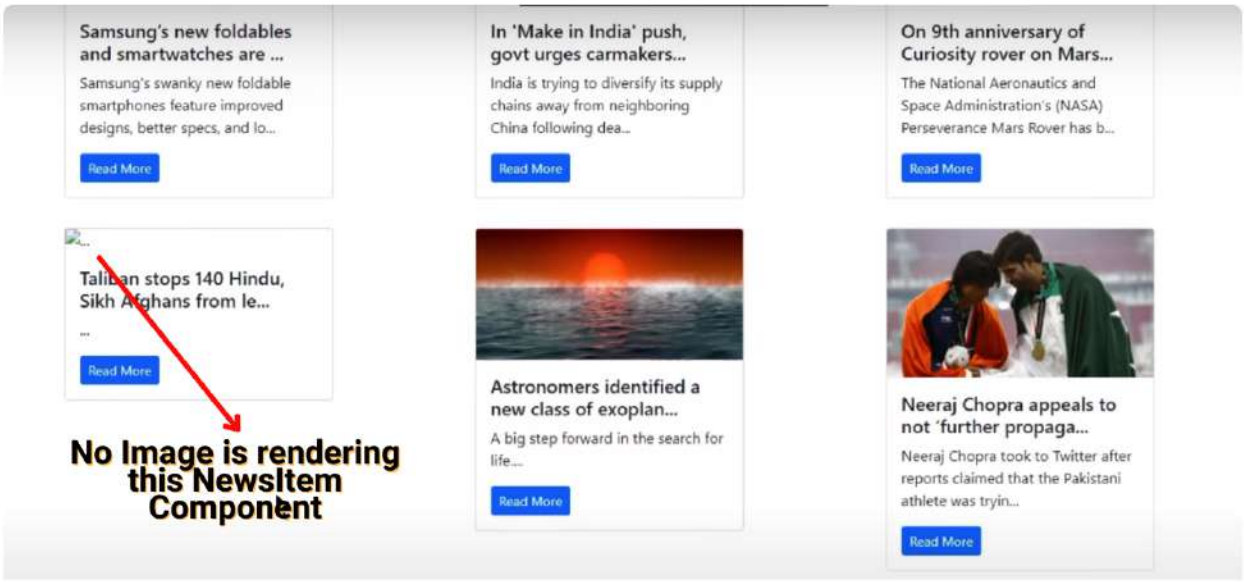

Figure 1.2: Image Not Rendering in the NewsItem

Here, the image is null for one of the Articles. Due to this reason, the image isn't rendering in Our NewsItem. To resolve this issue, we would use a default Image URL when the 'Url to the image of article' is Null. For that, we would use the ternary operators in the 'imgsrc' tag of NewsItem.js in the following way:

Code:

```
<img src={!imageUrl?"https://fdn.gsmarena.com/imgroot/news/21/08/xiaomi-smart-home-in
```

Hence, the 'image not rendering in the News Component' issue has been **successfully** resolved.