Adding Infinite Scroll to NewsMonkey | Complete React Course in Hindi #35

In an earlier video, we have learned about the component life cycle method of React. In this article, we would add an infinite scroll to the News Monkey application. So, without further ado let's begin:

Removing Previous and Next Button

Firstly, we will remove the Next/Previous button and the function assigned to them, as instead of those buttons we would be using the Infinite scroll to render all the available NewsItems on one page.

Installing react-infinite-scroll-component

To start using the Infinite scroll, we have to install the react-infinite-scroll-package npm package by using the below command:
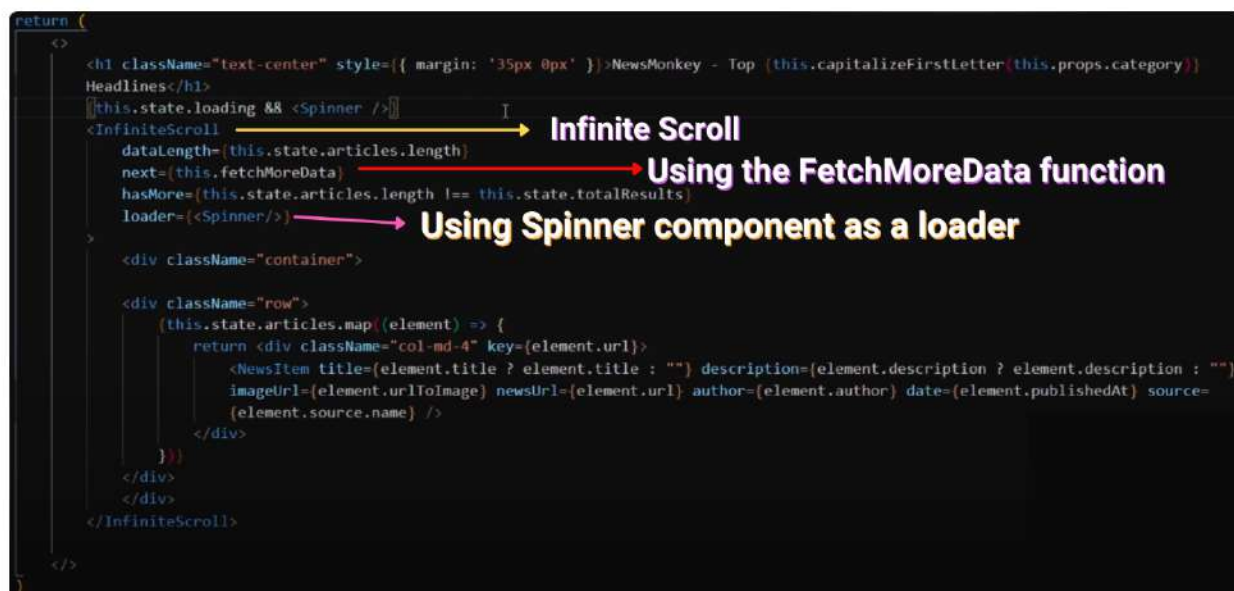
```
npm i react-infinite-scroll-component
```

In News.js:

Previously, we were returning a given set of NewsItems when the loading was false. As for now, we would concatenate more data in the already rendering data. That means, while scrolling down, more data will be added in the NewsComponent. To do so follow the below steps:

1. Adding Infinite Scroll

Firstly, make sure to import Infinite scroll and then add the infinite scroll in News Component as shown below:

```
return (
  <>
    <h1 className="text-center" style={{ margin: '35px 0px' }}>NewsMonkey - Top {this.capitalizeFirstLetter(this.props.category)}
    Headlines</h1>
    {this.state.loading && <Spinner />}
    <InfiniteScroll
      dataLength={this.state.articles.length}
      next={this.fetchMoreData}
      hasMore={this.state.articles.length !== this.state.totalResults}
      loader={<Spinner/>}
    >
      <div className="container">

        <div className="row">
          {this.state.articles.map((element) => {
            return <div className="col-md-4" key={element.url}>
              <NewsItem title={element.title ? element.title : ""} description={element.description ? element.description : ""}
              imageUrl={element.urlToImage} newsUrl={element.url} author={element.author} date={element.publishedAt} source=
              {element.source.name} />
            </div>
          })}
        </div>
      </div>
    </InfiniteScroll>

  </>
)
```

Infinite Scroll → `<InfiniteScroll`

Using the FetchMoreData function → `next={this.fetchMoreData}`

Using Spinner component as a loader → `loader={<Spinner/>}`

Figure 1.1: Using Infinite Scroll

Explanation: Above, we have set the data length equal to the article length. We would be creating a "fetchMoreData" function later on. After that, to check if more articles are available for rendering in the NewsComponent we have used "hasMore". Inside the constructor, remember to set the total results equal to 0. Finally, we have added the spinner component in the Loader.

2. Creating fetchMoreData Function

We have to write a function to render more NewsItem. Here's the code of the fetchMoreData function:

```
fetchMoreData = async () => {
    this.setState({ page: this.state.page + 1 })
    const url = `https://newsapi.org/v2/top-headlines?country=${this.props.country}&category=
    let data = await fetch(url);
    let parsedData = await data.json()
    this.setState({
        articles: this.state.articles.concat(parsedData.articles),
        totalResults: parsedData.totalResults,
        loading: false,
    })
};
```

Explanation: In the fetchMoreData function, we have set the state of the page number and have fetched the previously used Data. After that, We have concatenated the "parsedData.articles" in articles.

Result: Hence, we have successfully added an Infinite Scroll bar to the application.

Figure 1.2: NewsMonkey Application