Component Structure of our NewsMonkey React App | Complete React Course in Hindi #23

In the last video, we learned how to set up this project. In this project, we are going to create a news application named NewsMonkey. So, without further ado let's begin:

The component structure of our application

Structure: We would create a Navbar component at the top and at the middle of our application, we would add a 'news' component that would contain our news items.

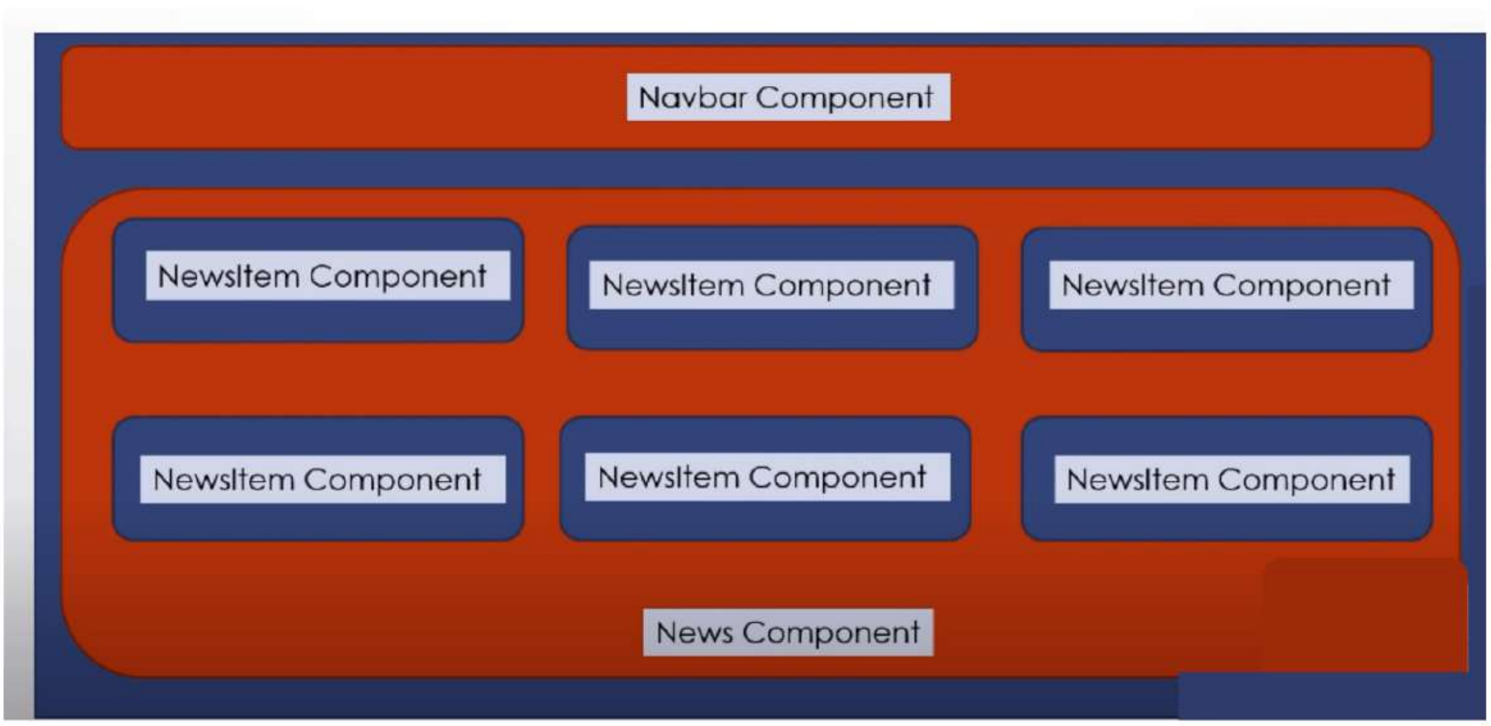


Figure 1.1: Structure of News Monkey

Navbar Component:

It will have navigation of different pages of our application, like About, Home, etc pages.

News Component:

The big red component is the News component. It will contain a lot of 'NewsItem' components.

NewsItem Component:

Many of these items will be specific news. For example Weather news, Sports News, Politics news, etc.

News Detail Component:

I would like to point out that later on, we would create a 'NewsDetail' Component. This component will show details of specific news when the reader clicks on a specific NewsItem. Our navbar will remain intact at the top of the application.

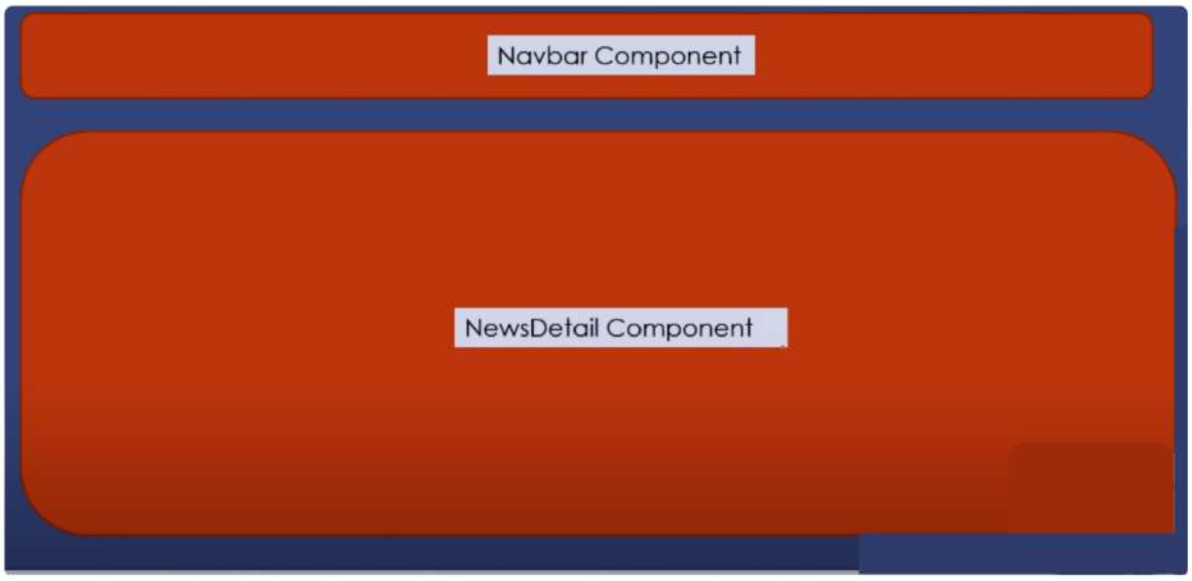


Figure 1.2: News Detail Component

Benefits: Structuring our app in this way lets us easily manage our application and also helps in reusing the components again and again.

Getting Started With NewsMonkey

Let’s Start our development server by using the ‘npm run start’ and start building our Amazing NewsMonkey Application:

Note: We will be using Bootstrap to get some design Components for our application.

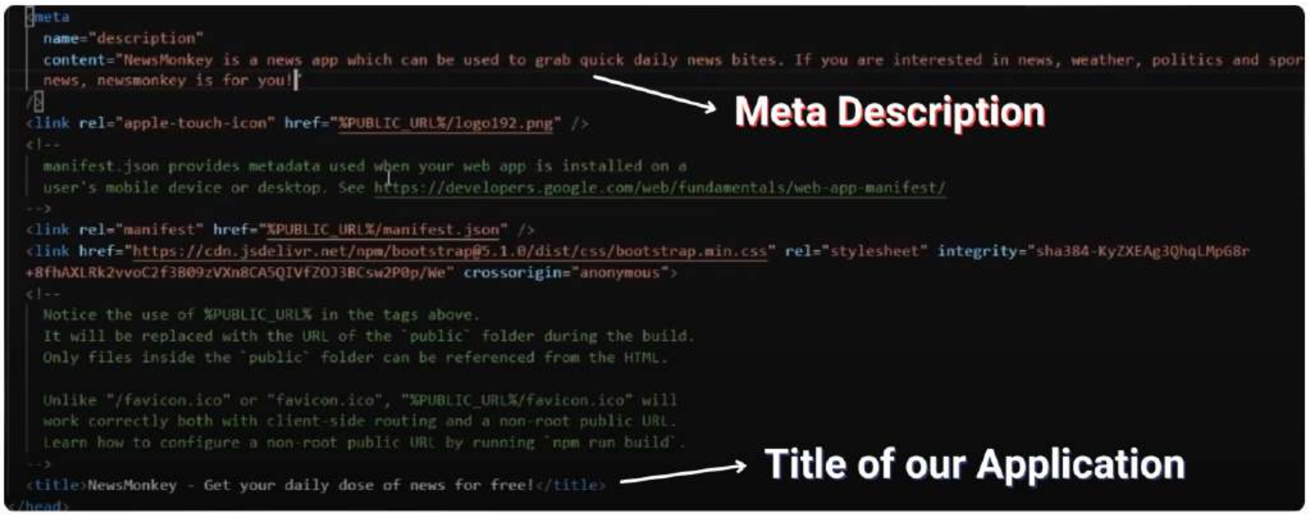
Using Bootstrap

Visit getBootstrap.com and copy-paste the starter Template of Bootstrap in your "Index.html" file. Now, we are ready to use the components of Bootstrap. Make sure to remember the below points while copying the code from Bootstrap:

1. Close those tags which don’t have a closing tag
2. Replace the "Class" keyword with "ClassName"
3. Replace href= “#” with href= “/”

Changing Title and meta description

We have added a fancy and SEO-friendly title and meta description to our NewsMonkey application:



```html
<meta
  name="description"
  content="NewsMonkey is a news app which can be used to grab quick daily news bites. If you are interested in news, weather, politics and spor
  news, newsmonkey is for you!"
/>
<link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
<!--
  manifest.json provides metadata used when your web app is installed on a
  user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
-->
<link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KyZXEAg3QhqLMpG8r
+8fhAXLRk2vvoC2f3B09zVXn8CA5QIVfZOJ3BCsw2P0p/We" crossorigin="anonymous">
<!--
  Notice the use of %PUBLIC_URL% in the tags above.
  It will be replaced with the URL of the `public` folder during the build.
  Only files inside the `public` folder can be referenced from the HTML.

  Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
  work correctly both with client-side routing and a non-root public URL.
  Learn how to configure a non-root public URL by running `npm run build`.
-->
<title>NewsMonkey - Get your daily dose of news for free!</title>
</head>
```

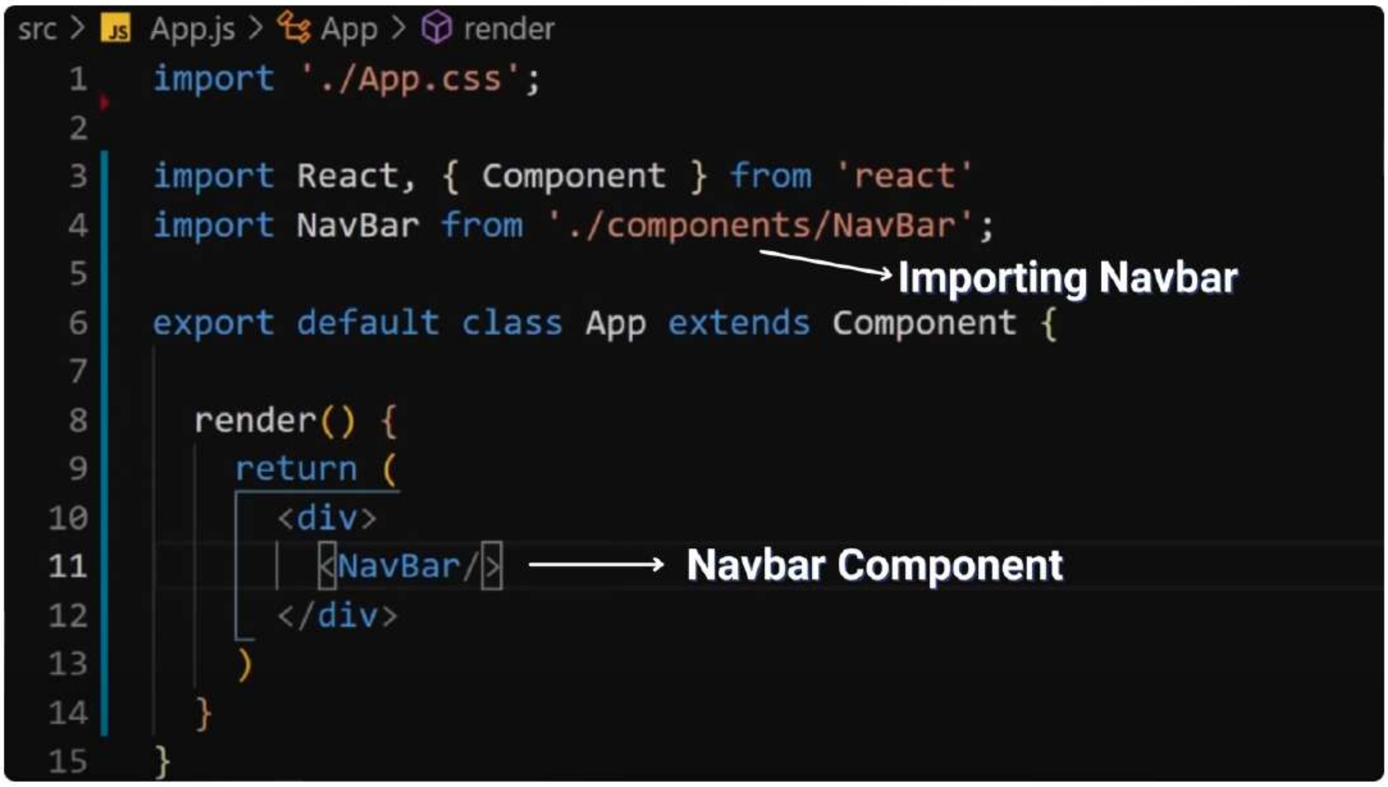
Figure 1.3: Adding title and meta description

Navbar component

Let’s begin creating our application from Navbar. Firstly we would create a "Navbar.js" file and would add a class-based component to it. Secondly, we would copy-paste the code of the Navbar component from Bootstrap. After this, we would make our desired changes in the Navbar. We have successfully created our React Component and now we would like to render it in our application.

Using Navbar component

We know that "App.js" is the file that is being rendered in our application. So, We have to use this Component in our "App.js" to render it in our application.



```
import './App.css';

import React, { Component } from 'react'
import NavBar from './components/NavBar';

export default class App extends Component {

  render() {
    return (
      <div>
        <NavBar/>
      </div>
    )
  }
}
```

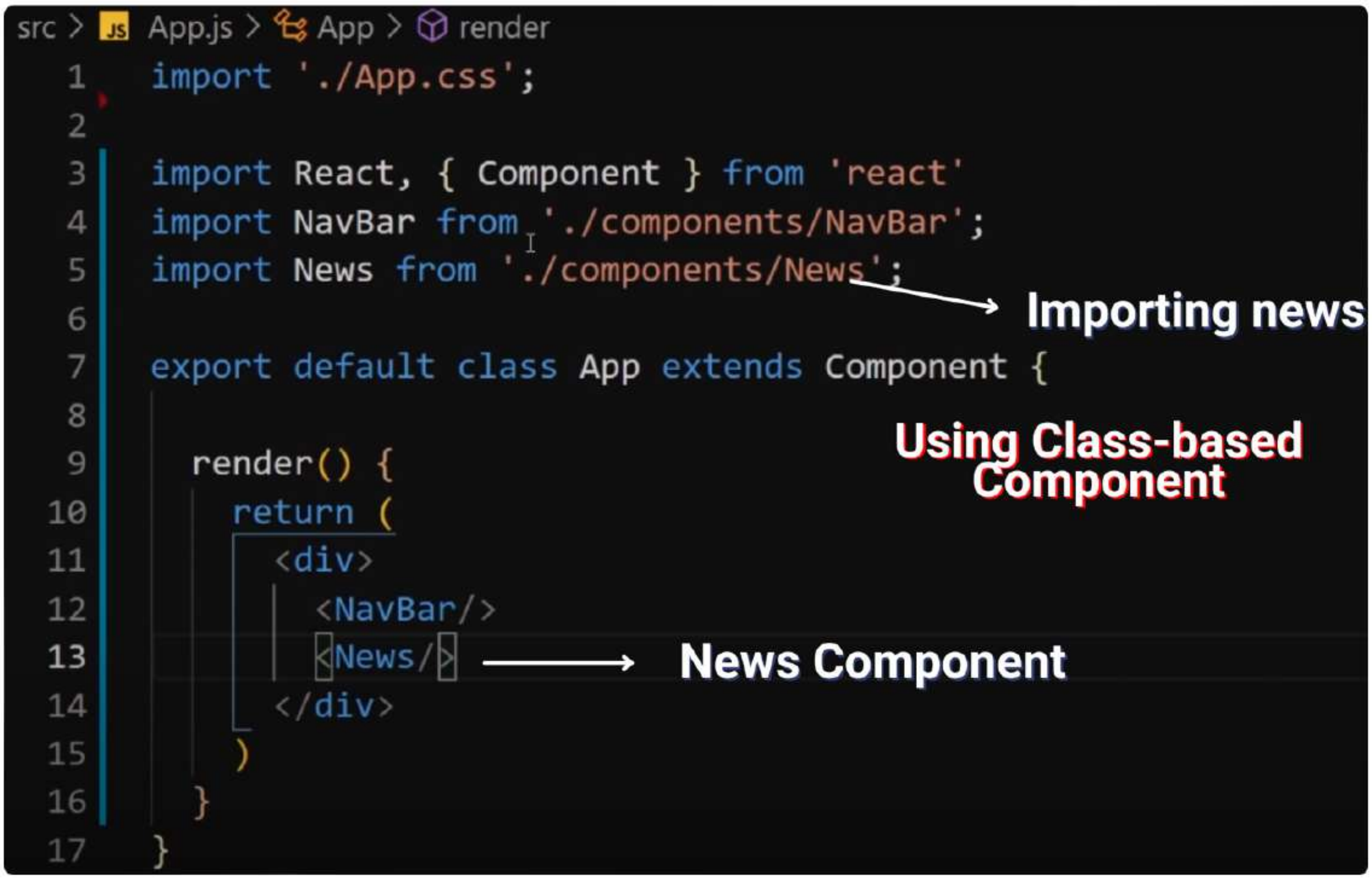
Figure 1.4: Using Navbar component

News Component

We would like to create a ‘news’ component, that would reside in the center of our application, and it will contain all the NewsItem components. So let’s create a new file:

News.js:

Create a "news.js" file and add a Class-based component to it. Now, we would return the desired content which we want to render in our application. After doing so, we would add this component to our "App.js".

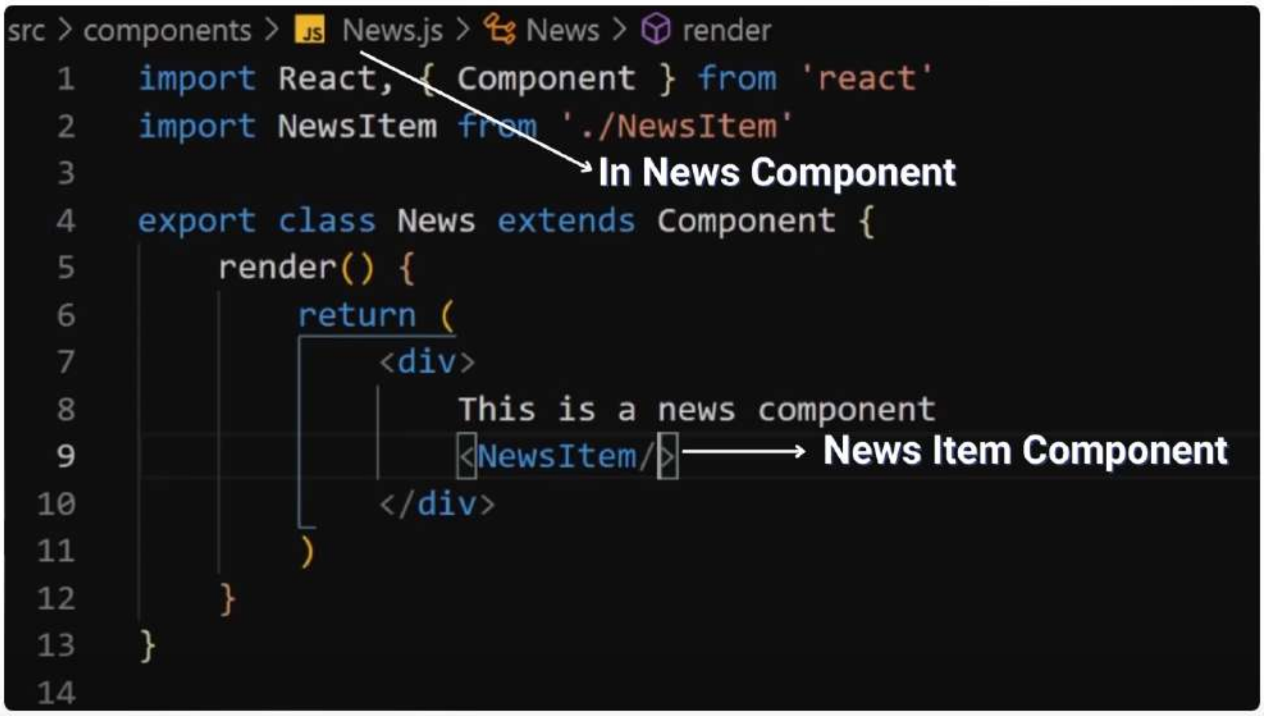


```
import './App.css';

import React, { Component } from 'react'
import NavBar from './components/NavBar';
import News from './components/News';

export default class App extends Component {

  render() {
    return (
      <div>
        <NavBar/>
        <News/>
      </div>
    )
  }
}
```

Figure 1.5: Using News Component

"NewsItem.js"

Create a "NewsItem.js" file and add a Class-based component to it. Now, we would return the desired content which we want to render in our application. After doing so, we would add this component to our "News.js". Hence, Our News.js file is being rendered in app.js and the News.js file contains our NewsItem Component.



```
import React, { Component } from 'react'
import NewsItem from './NewsItem'

export class News extends Component {
    render() {
        return (
            <div>
                This is a news component
                <NewsItem/>
            </div>
        )
    }
}

```