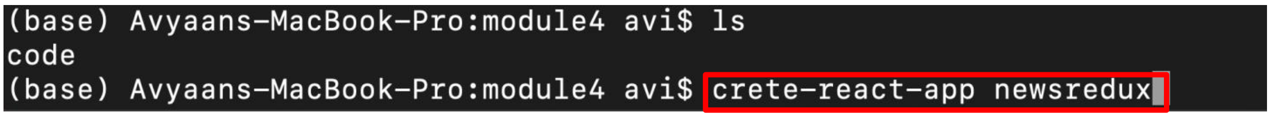
Module 4: React State Management using Redux

Demo Document 2

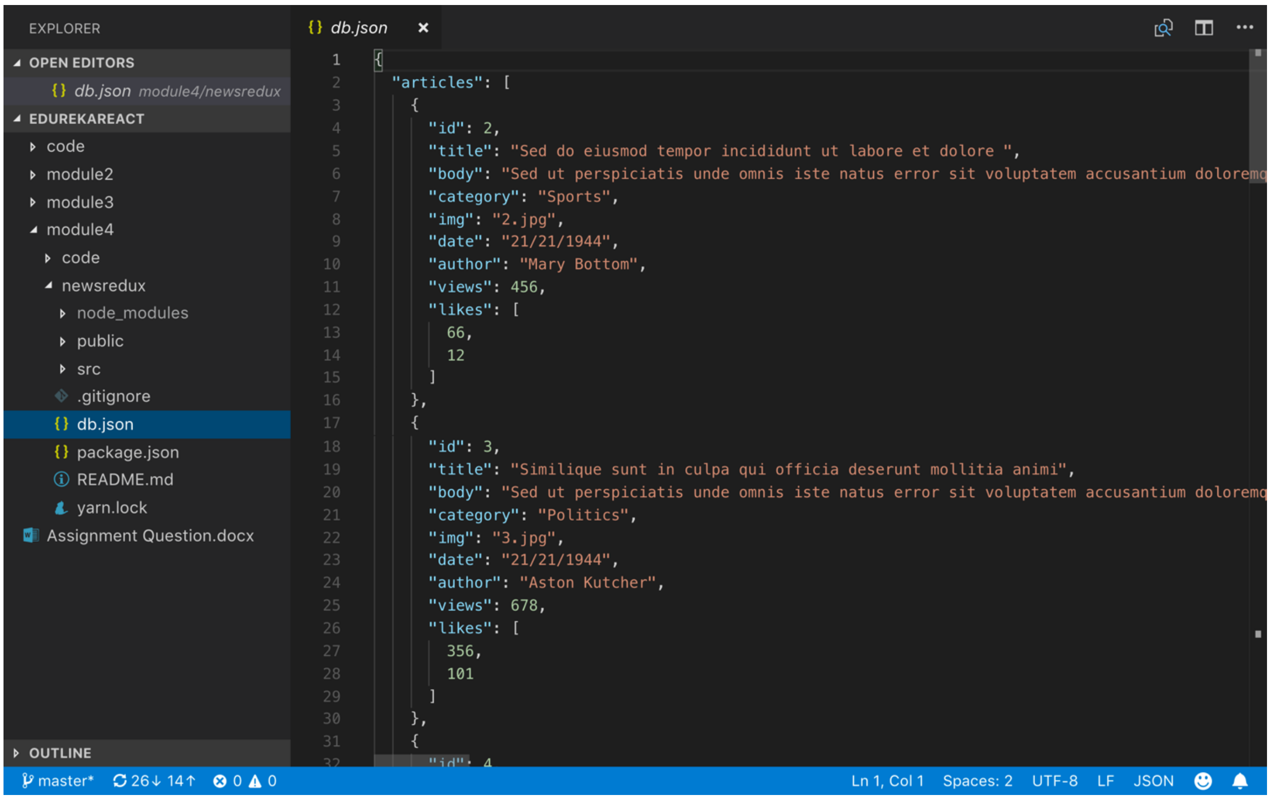
To Build a News Application using React and Redux

Step 1: To create a news application using React and Redux, first create a seed of react using ***create-react-app <application_name>.***



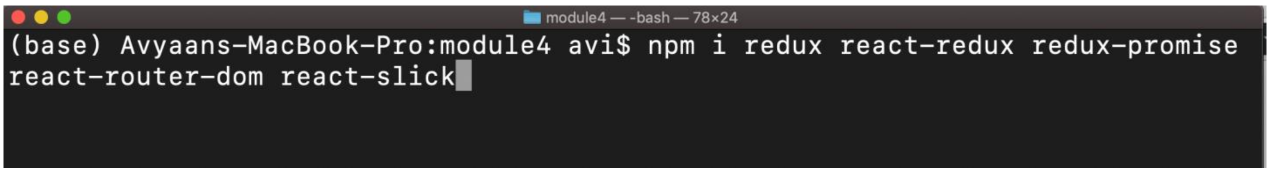
```
(base) Avyaans-MacBook-Pro:module4 avi$ ls
code
(base) Avyaans-MacBook-Pro:module4 avi$ crete-react-app newsredux
```

Step 2: Create a db.json file, this file should conatin the JSON data related to Articles and Gallery.



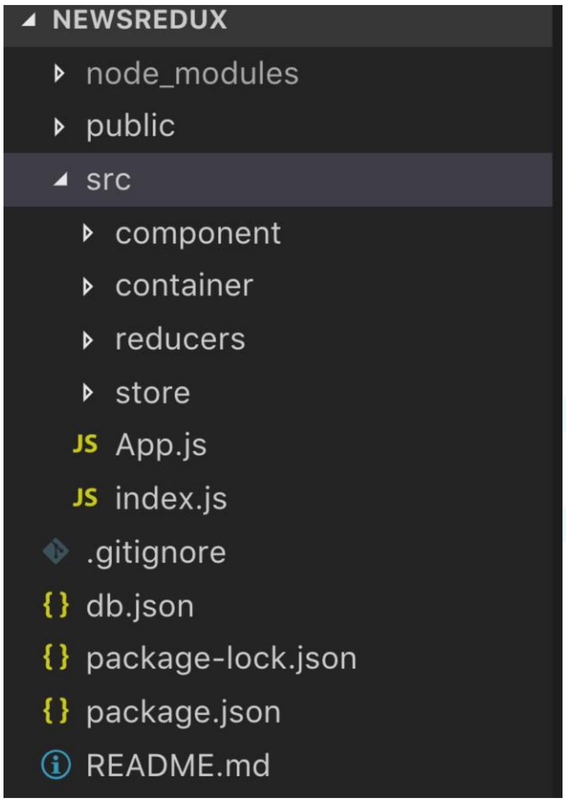
```
{
  "articles": [
    {
      "id": 2,
      "title": "Sed do eiusmod tempor incididunt ut labore et dolore ",
      "body": "Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremq
      "category": "Sports",
      "img": "2.jpg",
      "date": "21/21/1944",
      "author": "Mary Bottom",
      "views": 456,
      "likes": [
        66,
        12
      ]
    },
    {
      "id": 3,
      "title": "Similique sunt in culpa qui officia deserunt mollitia animi",
      "body": "Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremq
      "category": "Politics",
      "img": "3.jpg",
      "date": "21/21/1944",
      "author": "Aston Kutcher",
      "views": 678,
      "likes": [
        356,
        101
      ]
    },
    {
      "id": 4
```

Step 3: Install packages: redux, redux-promise (for middleware), react-slick (for slider), react-router-dom (for routing) and react-redux.



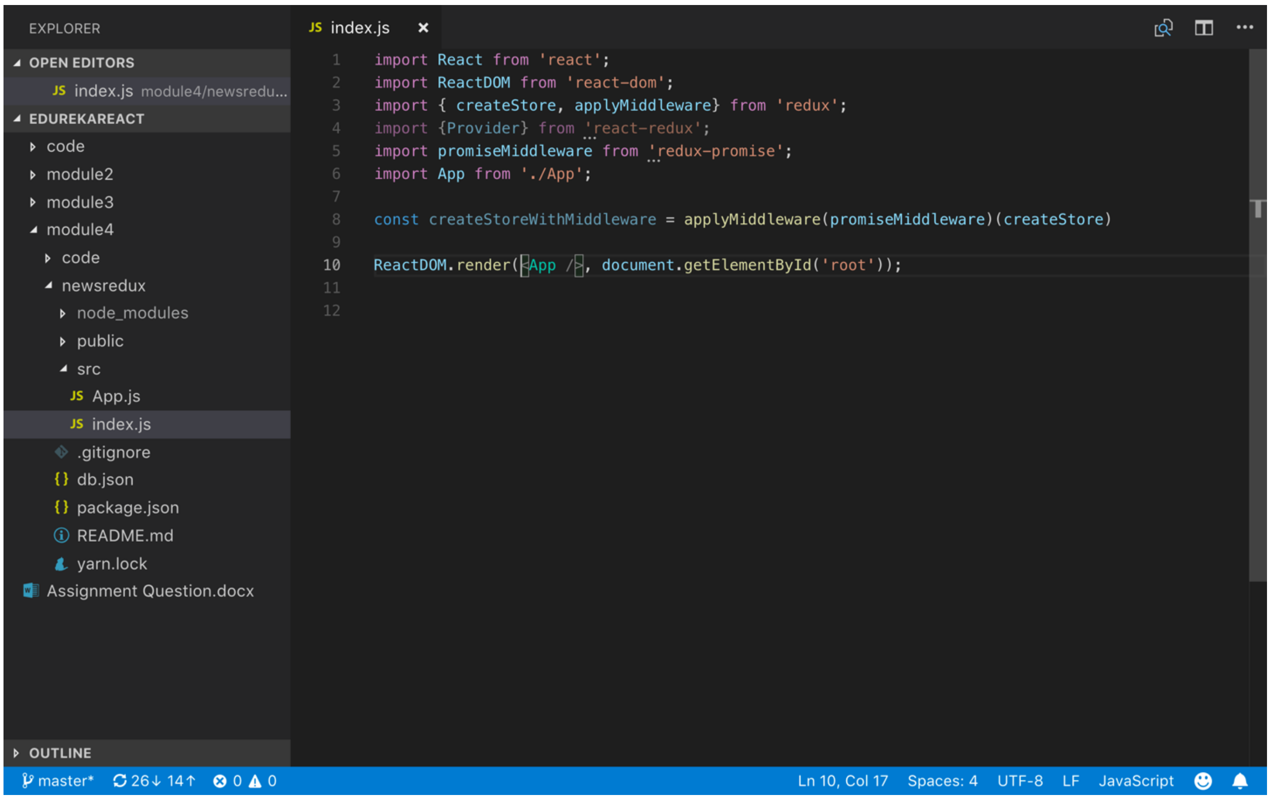
```
(base) Avyaans-MacBook-Pro:module4 avi$ npm i redux react-redux redux-promise
react-router-dom react-slick
```

Step 4: Create the folder structure as shown below:



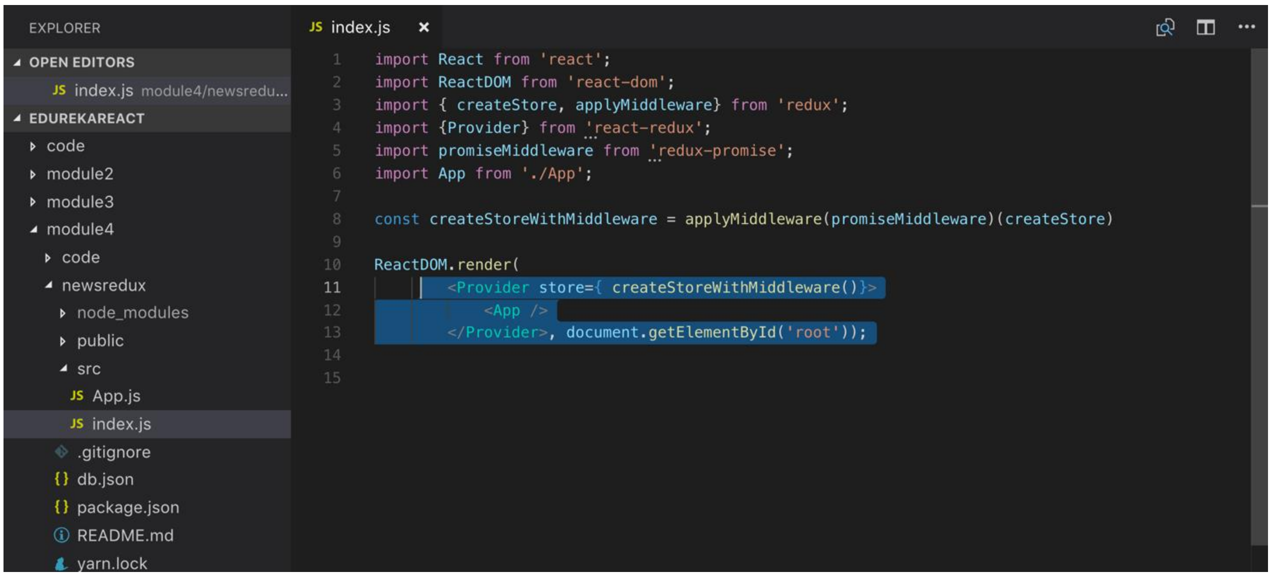
```
NEWSREDUX
  node_modules
  public
  src
    component
    container
    reducers
    store
    App.js
    index.js
  .gitignore
  db.json
  package-lock.json
  package.json
  README.md
```

Step 5: In the index.js file, create a store by importing ***createStore*** and ***applyMiddleware*** methods.



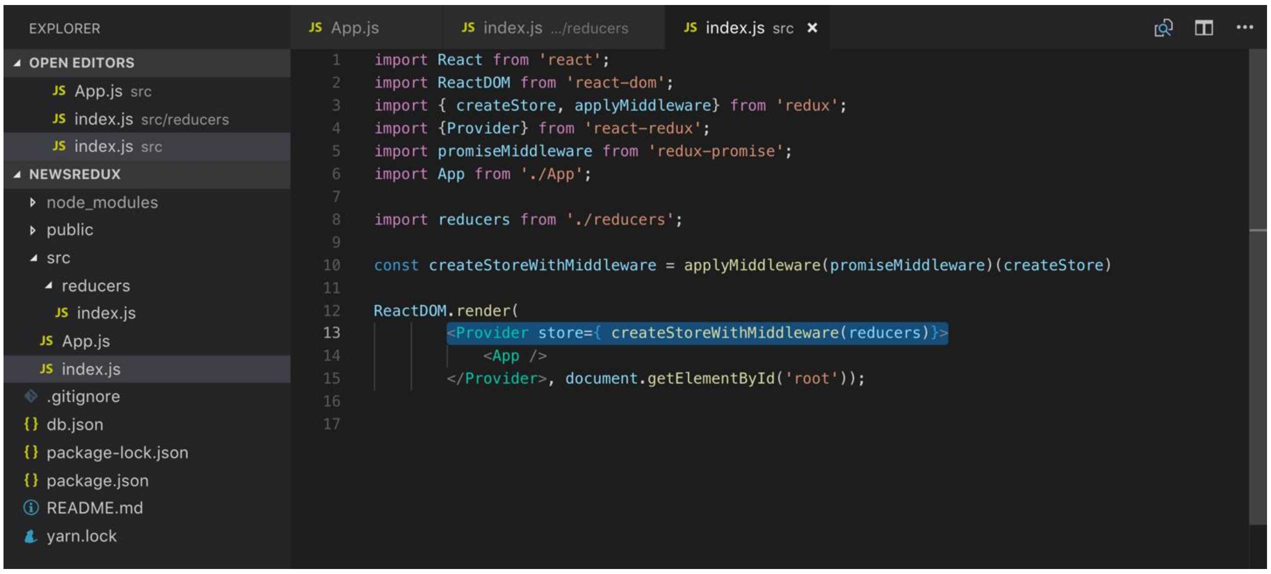
```
import React from 'react';
import ReactDOM from 'react-dom';
import { createStore, applyMiddleware} from 'redux';
import {Provider} from 'react-redux';
import promiseMiddleware from 'redux-promise';
import App from './App';

const createStoreWithMiddleware = applyMiddleware(promiseMiddleware)(createStore)

ReactDOM.render(<App />, document.getElementById('root'));
```

Step 6: Now import the ***Provider*** and wrap "App file" inside the Provider.



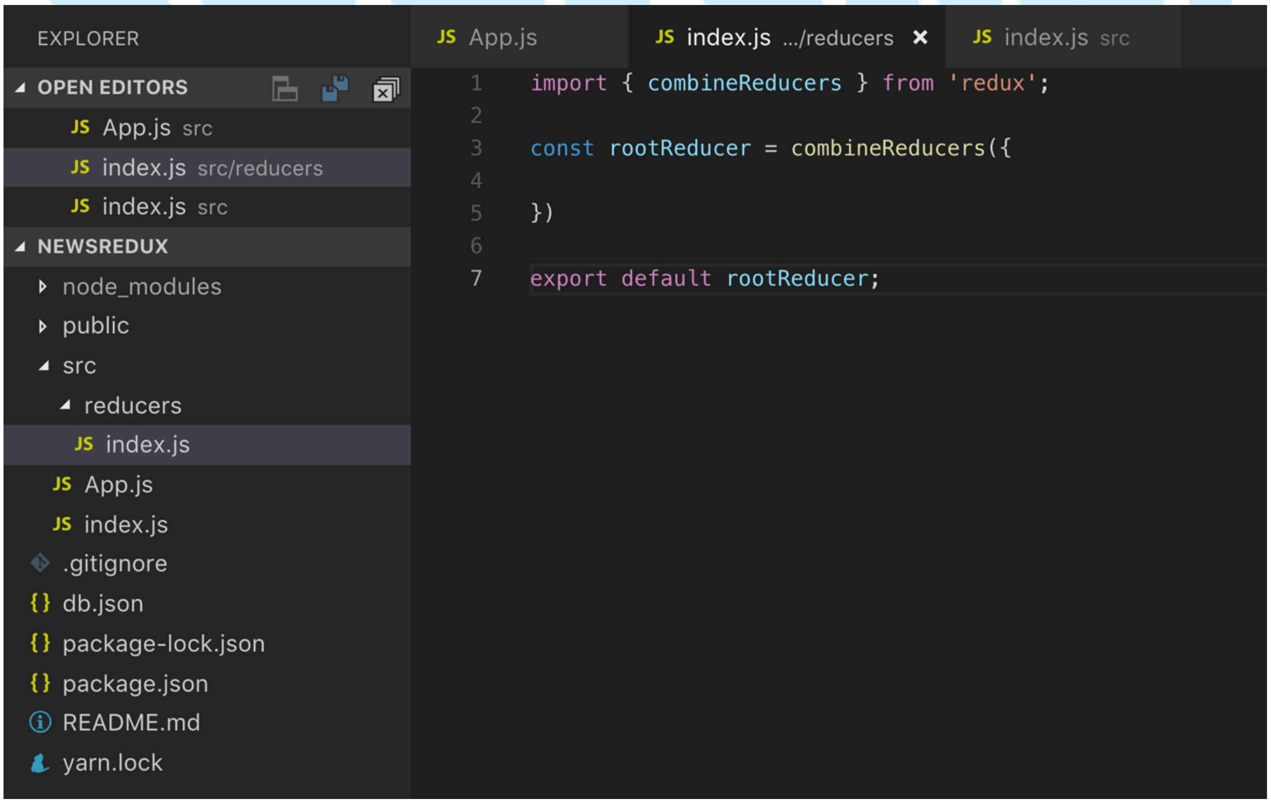
```
import React from 'react';
import ReactDOM from 'react-dom';
import { createStore, applyMiddleware} from 'redux';
import {Provider} from 'react-redux';
import promiseMiddleware from 'redux-promise';
import App from './App';

const createStoreWithMiddleware = applyMiddleware(promiseMiddleware)(createStore)

ReactDOM.render(
    <Provider store={ createStoreWithMiddleware()}>
        <App />
    </Provider>, document.getElementById('root'));
```

Step 7: Connect the ***store*** and ***reducer*** by passing the reducer (as a parameter) to store within the Provider.



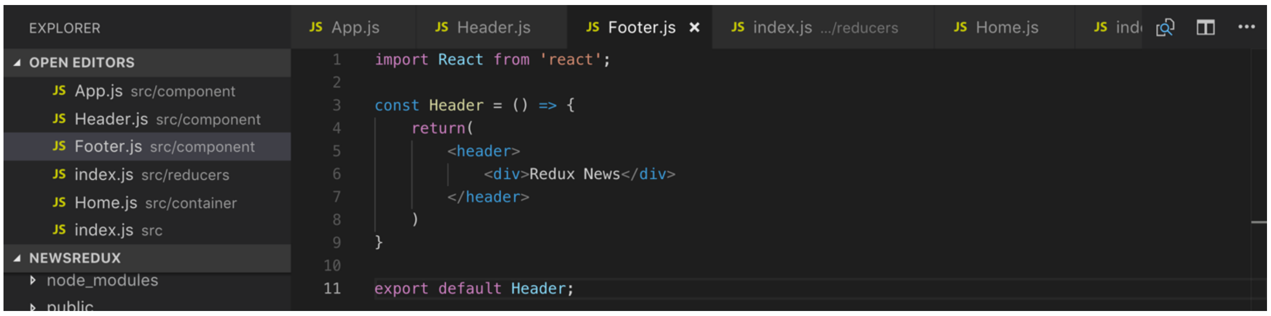
```
import React from 'react';
import ReactDOM from 'react-dom';
import { createStore, applyMiddleware} from 'redux';
import {Provider} from 'react-redux';
import promiseMiddleware from 'redux-promise';
import App from './App';

import reducers from './reducers';

const createStoreWithMiddleware = applyMiddleware(promiseMiddleware)(createStore)

ReactDOM.render(
        <Provider store={ createStoreWithMiddleware(reducers)}>
            <App />
        </Provider>, document.getElementById('root'));
```

Step 8: In the ***reducers folder*** create index.js file and add ***combineReducers*** which will be acting like a ***rootReducers***.



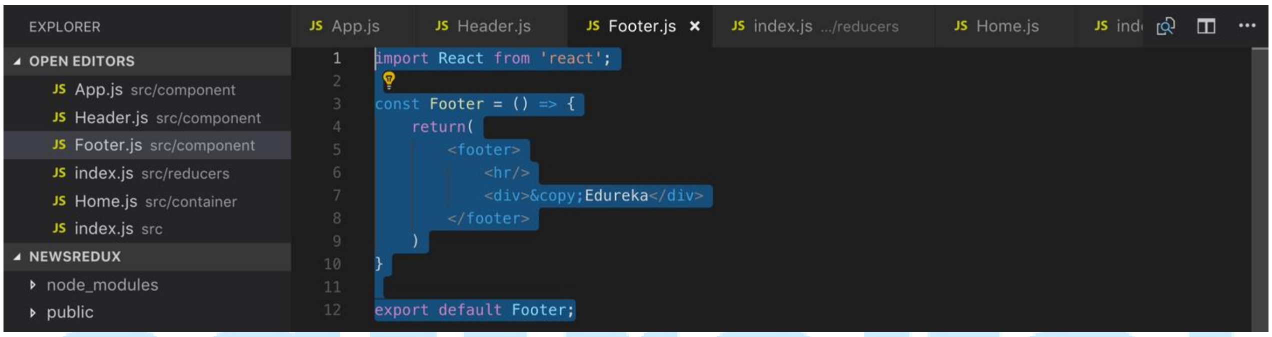
```
import { combineReducers } from 'redux';

const rootReducer = combineReducers({

})

export default rootReducer;
```

Step 9: Create the Header.js file using functional component as shown below:



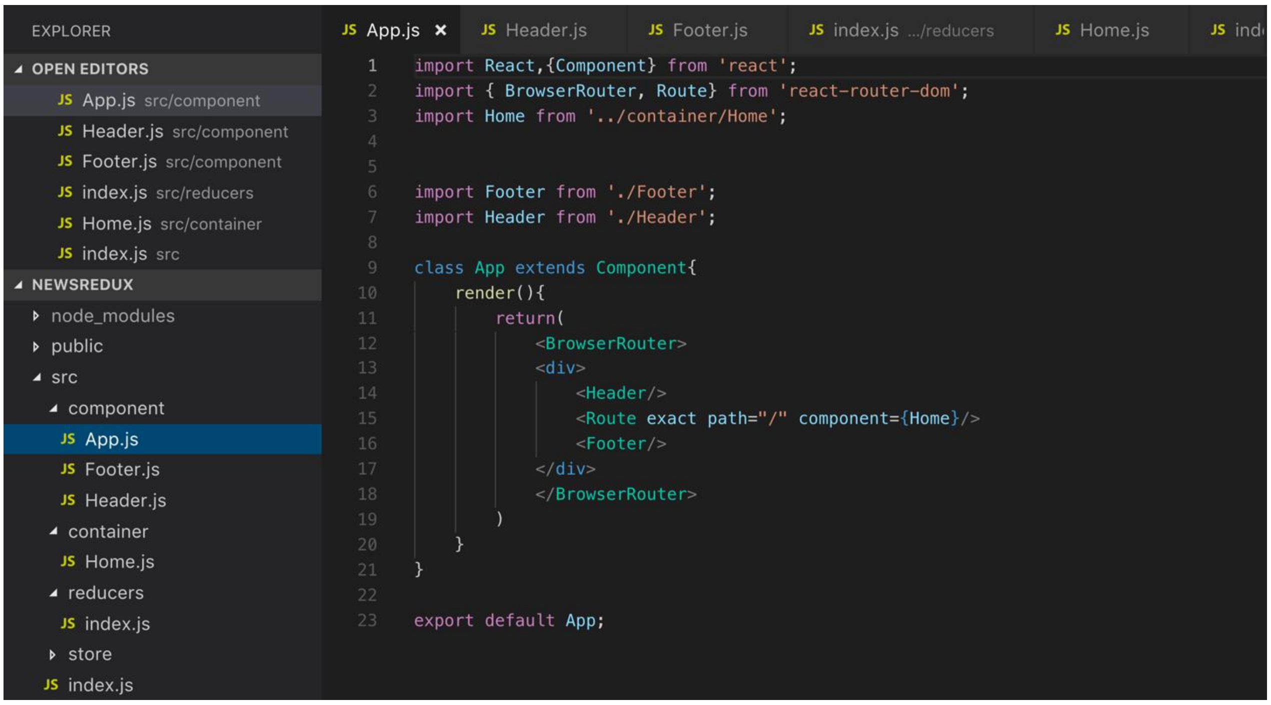
```
import React from 'react';

const Header = () => {
    return(
        <header>
            <div>Redux News</div>
        </header>
    )
}

export default Header;
```

Step 10: Similarly create a Footer.js file



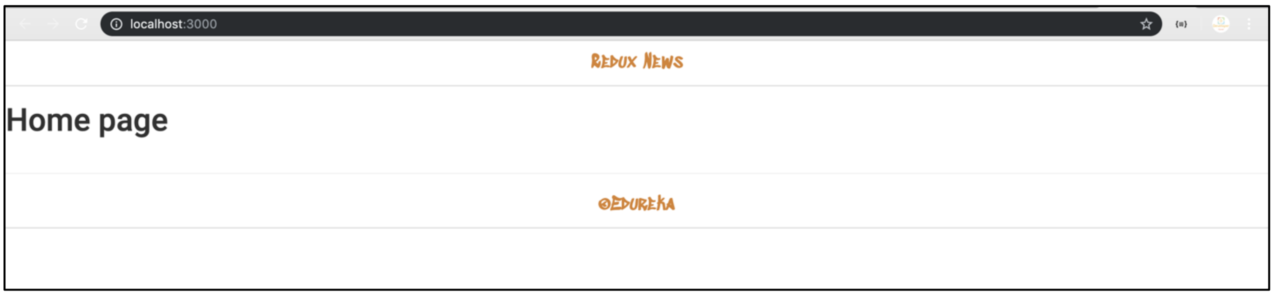
```
import React from 'react';

const Footer = () => {
    return(
        <footer>
            <hr/>
            <div>&copy;Edureka</div>
        </footer>
    )
}

export default Footer;
```

Step 11: In the "App.js" file add routing. Import ***header*** and ***footer*** files, also add default route as home page.

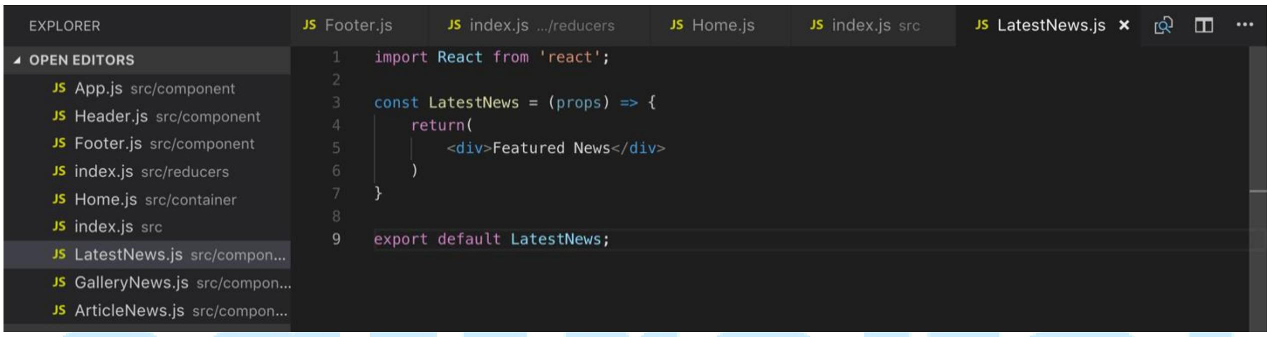


```
import React,{Component} from 'react';
import { BrowserRouter, Route} from 'react-router-dom';
import Home from '../container/Home';


import Footer from './Footer';
import Header from './Header';

class App extends Component{
    render(){
        return(
            <BrowserRouter>
            <div>
                <Header/>
                <Route exact path="/" component={Home}/>
                <Footer/>
            </div>
            </BrowserRouter>
        )
    }
}

export default App;
```

Step 12: Check the output. (you can change the CSS for design and styling).

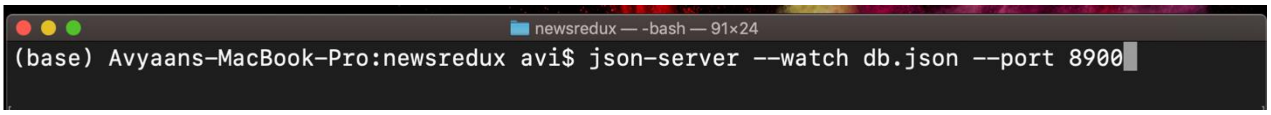


Step 13: Create a component for the latest news where we will load the data from an ***API.***



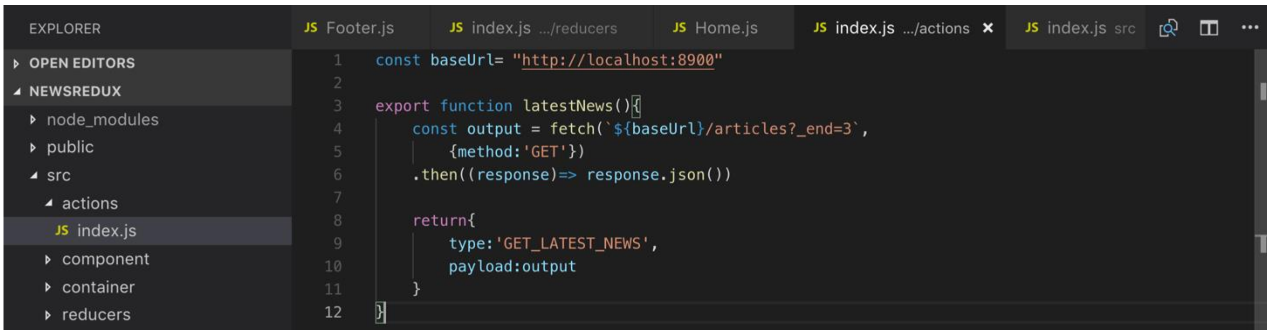
```
import React from 'react';

const LatestNews = (props) => {
    return(
        <div>Featured News</div>
    )
}

export default LatestNews;
```

Step 14: For the above step first API must be in running state. Open a new terminal and write the command: ***json-server --watch < JSON file holding JSON data> --port <number> ,*** to start the JSON server.



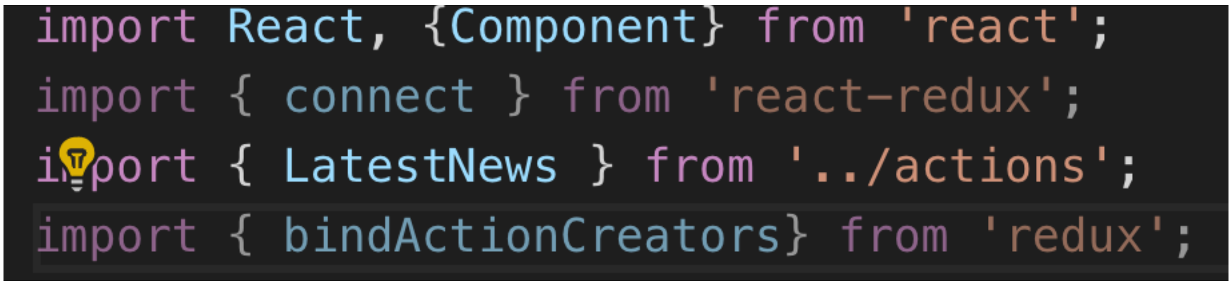
```
(base) Avyaans-MacBook-Pro:newsredux avi$ json-server --watch db.json --port 8900
```

Step 15: Create an “index.js” file within ***actions*** folder, here with the help of ***fetch***() method make a call to the API. With the help of function latestNews(), ***actions*** will return *type and payload.*



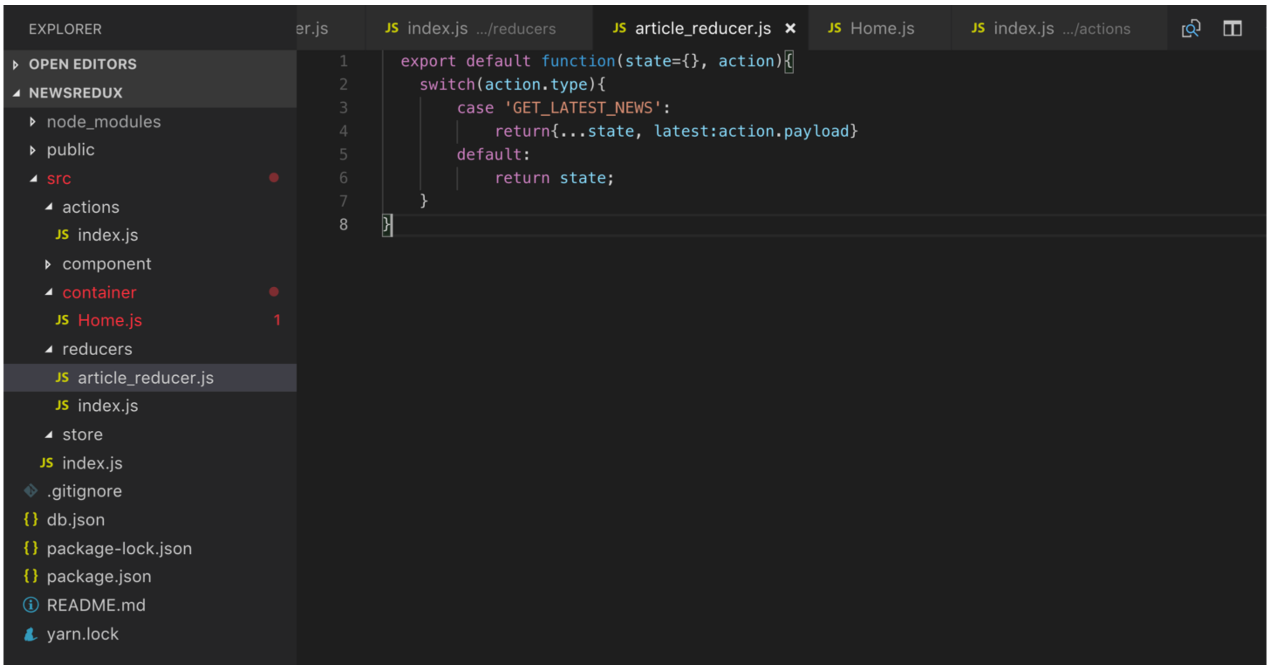
```
const baseUrl= "http://localhost:8900"

export function latestNews(){
    const output = fetch(`${baseUrl}/articles?_end=3`,
        {method:'GET'})
    .then((response)=> response.json())

    return{
        type:'GET_LATEST_NEWS',
        payload:output
    }
}
```

Step 16: In the home file, import ***connect***, ***action*** and ***bindActionCreators*** for the purpose of resolving API and binding the data.



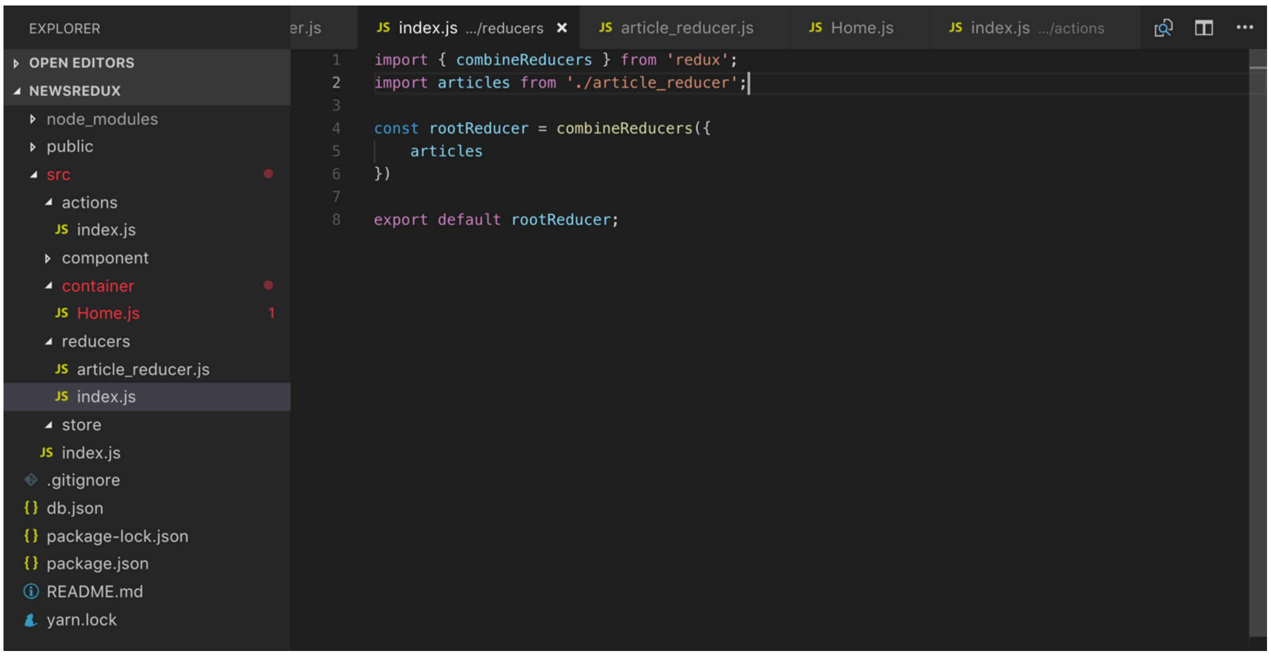
```
import React, {Component} from 'react';
import { connect } from 'react-redux';
import { LatestNews } from '../actions';
import { bindActionCreators} from 'redux';
```

Step 17: In reducers folder create ***articles_reducer.js*** file to perform state management and return state and payload.



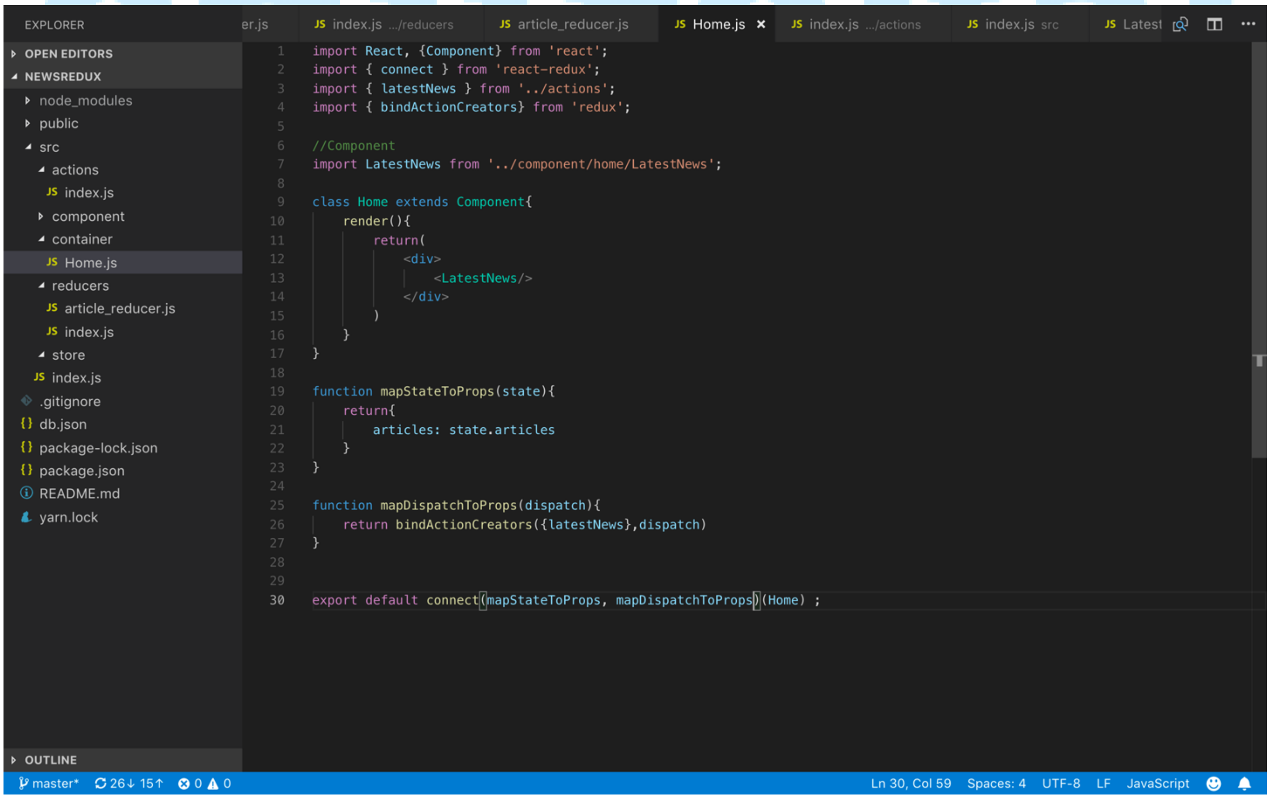
```
export default function(state={}, action){
  switch(action.type){
      case 'GET_LATEST_NEWS':
          return{...state, latest:action.payload}
      default:
          return state;
  }
}
```

Step 18: All the reducers should be mapped to the main reducers using ***combineReducer()*** method.



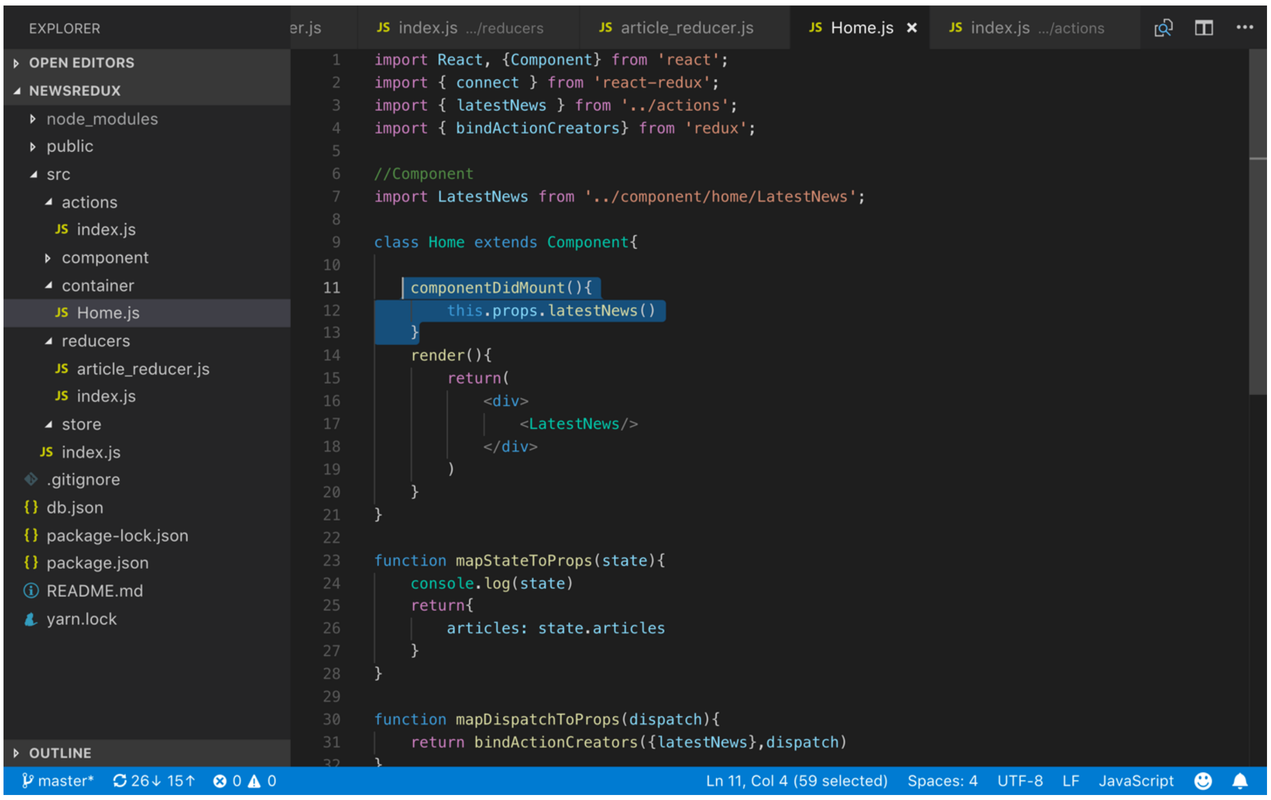
```
import { combineReducers } from 'redux';
import articles from './article_reducer';

const rootReducer = combineReducers({
    articles
})

export default rootReducer;
```

Step 19: Finally create two more methods, ***mapStateToProps()*** to recieve state and ***mapDispatchToProps()*** to bind the action.



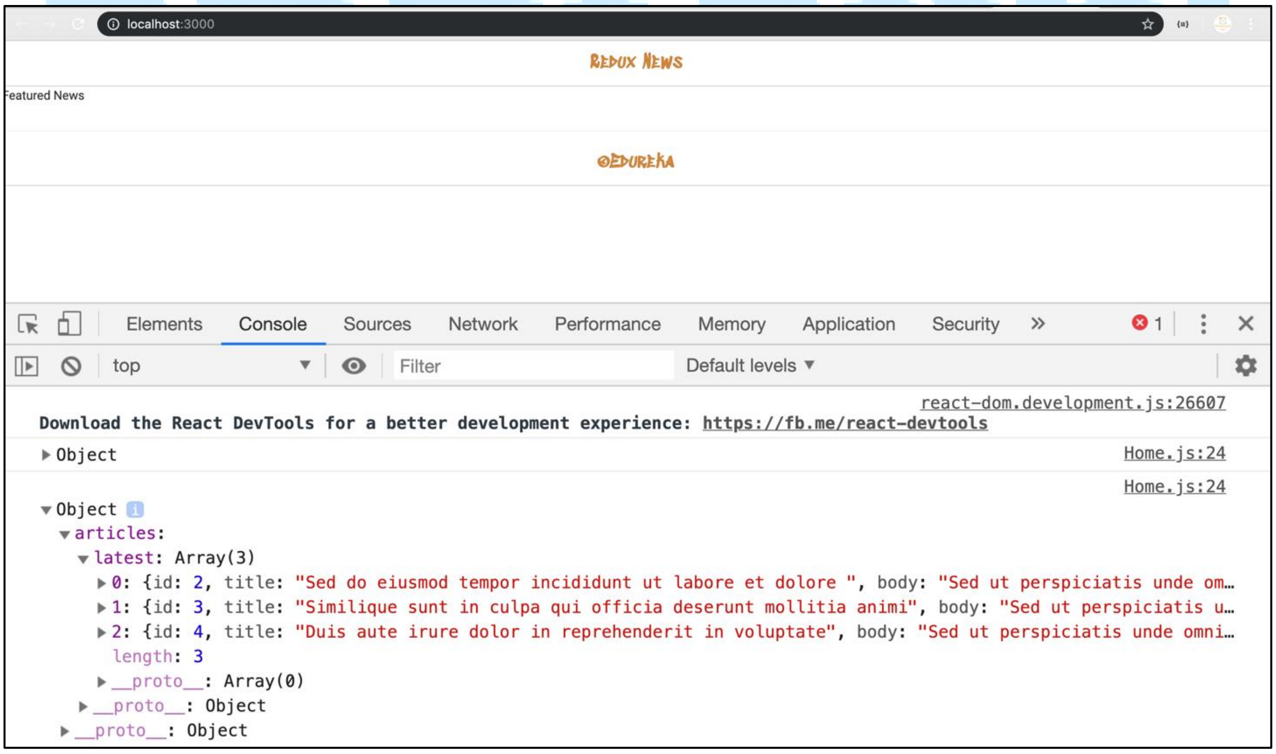
```
import React, {Component} from 'react';
import { connect } from 'react-redux';
import { latestNews } from '../actions';
import { bindActionCreators} from 'redux';

//Component
import LatestNews from '../component/home/LatestNews';

class Home extends Component{
    render(){
        return(
            <div>
                <LatestNews/>
            </div>
        )
    }
}

function mapStateToProps(state){
    return{
        articles: state.articles
    }
}

function mapDispatchToProps(dispatch){
    return bindActionCreators({latestNews},dispatch)
}


export default connect(mapStateToProps, mapDispatchToProps)(Home) ;
```

Step 20: With the help of ***lifecycle hook*** call the action and this will return the data in the state.



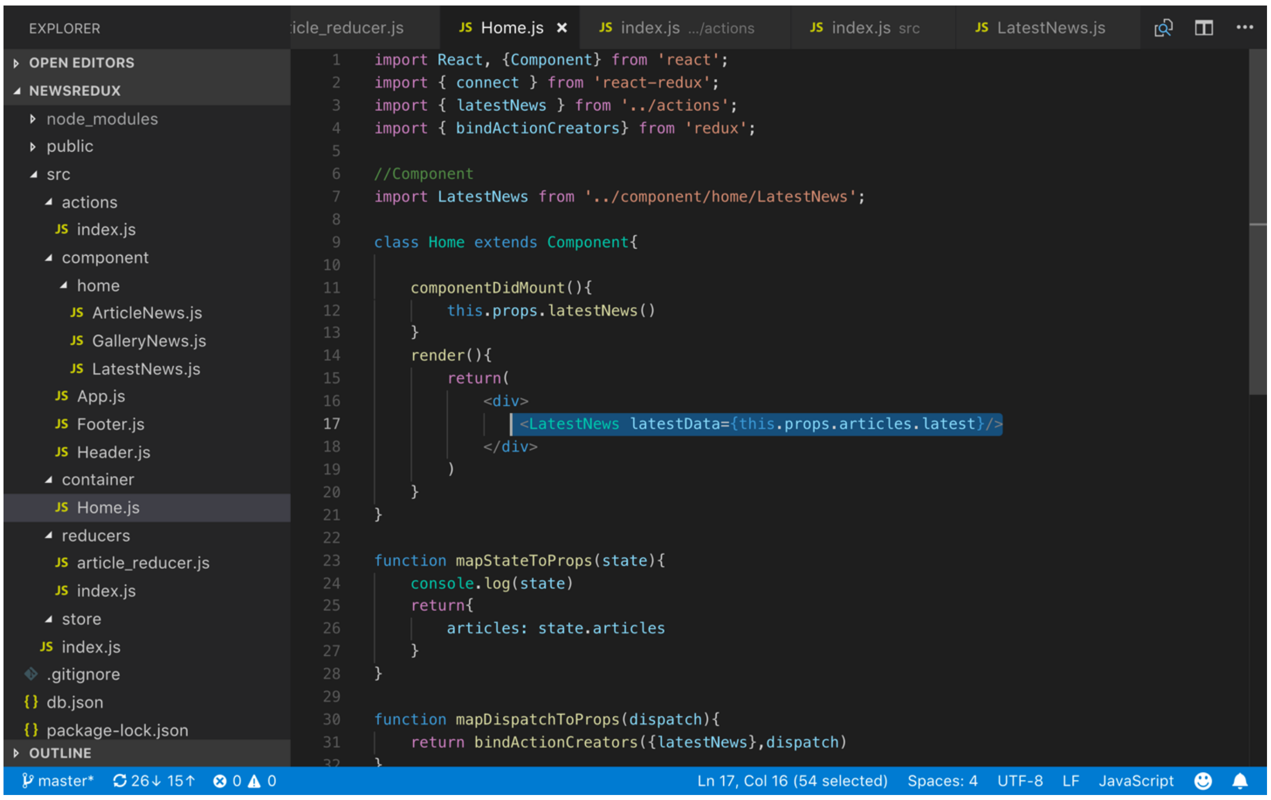
```
import React, {Component} from 'react';
import { connect } from 'react-redux';
import { latestNews } from '../actions';
import { bindActionCreators} from 'redux';

//Component
import LatestNews from '../component/home/LatestNews';

class Home extends Component{

    componentDidMount(){
        this.props.latestNews()
    }
    render(){
        return(
            <div>
                <LatestNews/>
            </div>
        )
    }
}

function mapStateToProps(state){
    console.log(state)
    return{
        articles: state.articles
    }
}

function mapDispatchToProps(dispatch){
    return bindActionCreators({latestNews},dispatch)
```

Step 21: Check the working of the actions in the browser console.



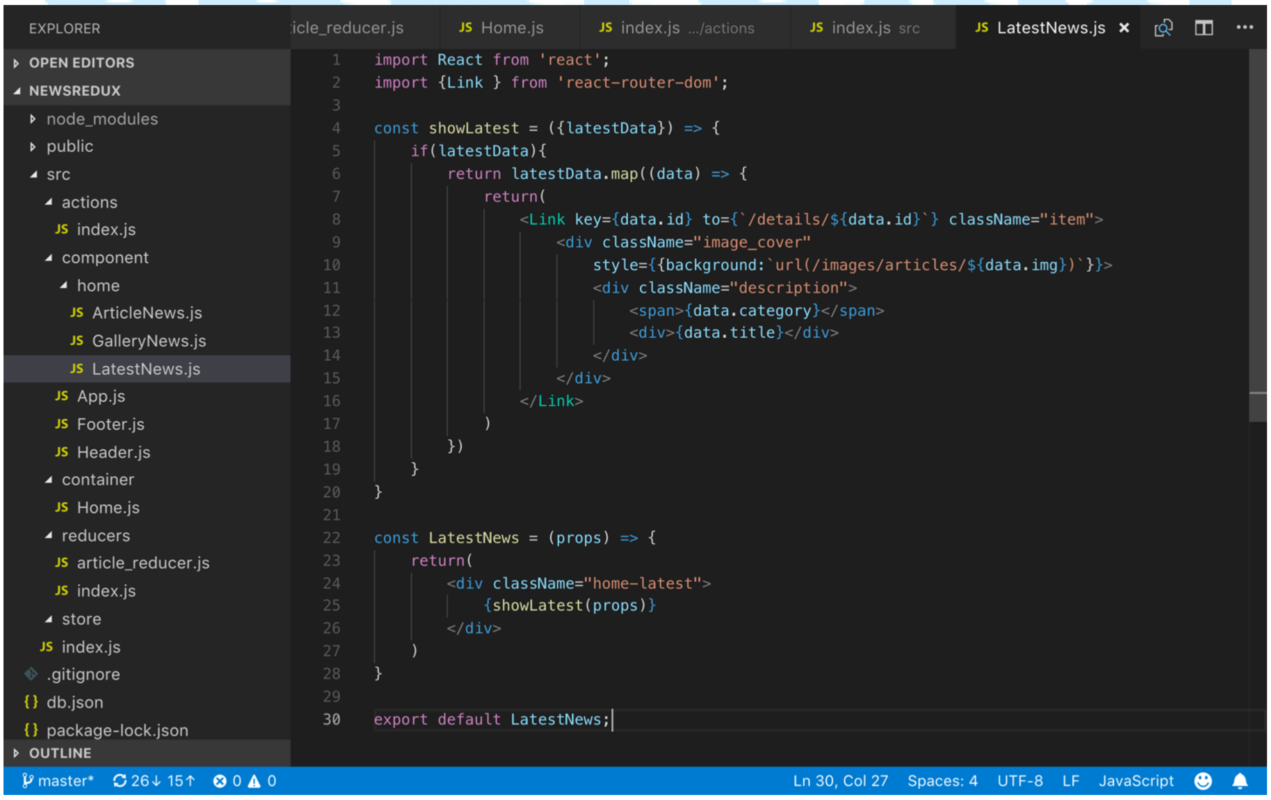
```
                                                   react-dom.development.js:26607
Download the React DevTools for a better development experience: https://fb.me/react-devtools
▶Object                                                               Home.js:24
                                                                      Home.js:24
▼Object
  ▼articles:
    ▼latest: Array(3)
      ▶0: {id: 2, title: "Sed do eiusmod tempor incididunt ut labore et dolore ", body: "Sed ut perspiciatis unde om…
      ▶1: {id: 3, title: "Similique sunt in culpa qui officia deserunt mollitia animi", body: "Sed ut perspiciatis u…
      ▶2: {id: 4, title: "Duis aute irure dolor in reprehenderit in voluptate", body: "Sed ut perspiciatis unde omni…
       length: 3
      ▶__proto__: Array(0)
    ▶__proto__: Object
  ▶__proto__: Object
```

Step 22: Send the received data to latestnews component, to render the data.



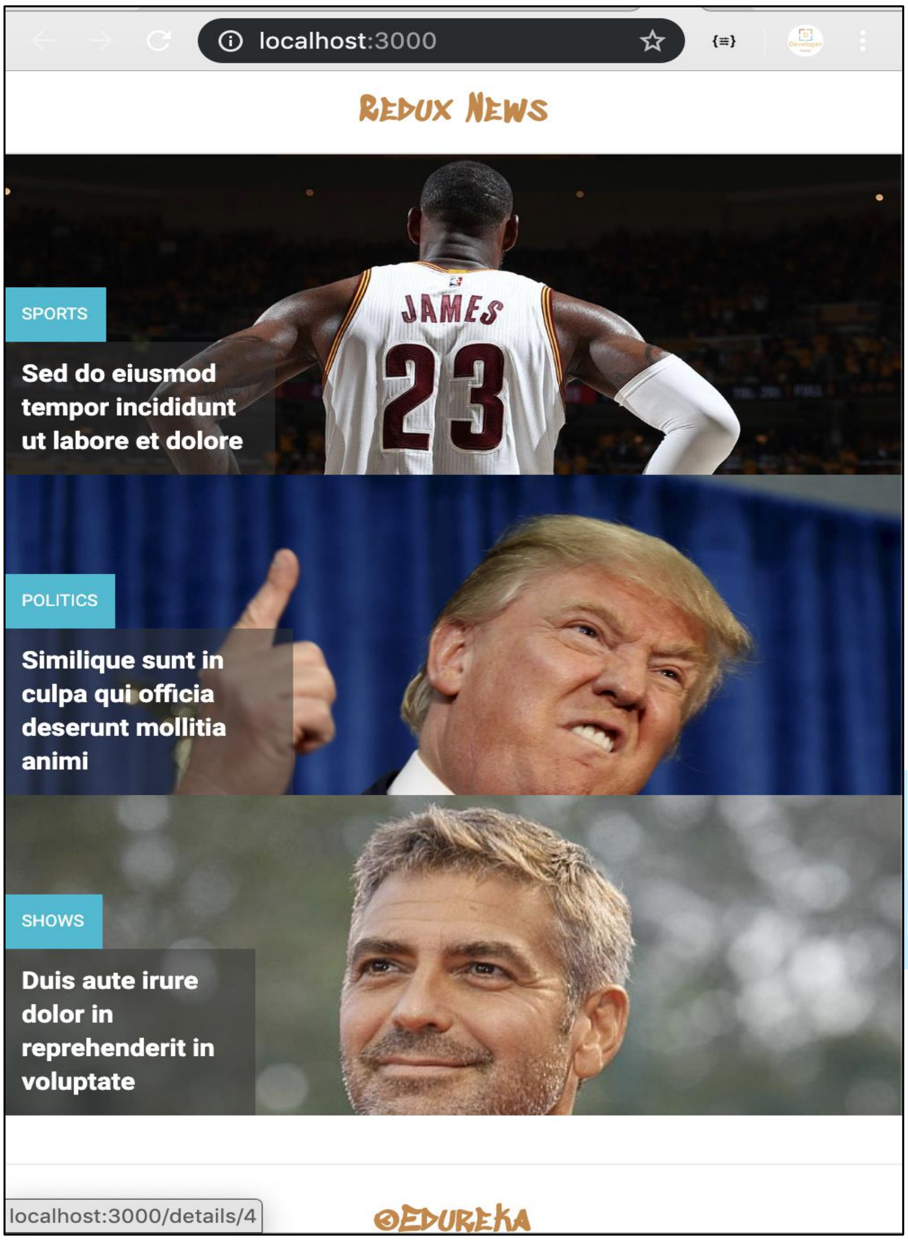
```
import React, {Component} from 'react';
import { connect } from 'react-redux';
import { latestNews } from '../actions';
import { bindActionCreators} from 'redux';

//Component
import LatestNews from '../component/home/LatestNews';

class Home extends Component{

    componentDidMount(){
        this.props.latestNews()
    }
    render(){
        return(
            <div>
                <LatestNews latestData={this.props.articles.latest}/>
            </div>
        )
    }
}

function mapStateToProps(state){
    console.log(state)
    return{
        articles: state.articles
    }
}

function mapDispatchToProps(dispatch){
    return bindActionCreators({latestNews},dispatch)
```

Step 23: In the latest News we will use MAP to iterate over the data and will list it as link.



```
import React from 'react';
import {Link } from 'react-router-dom';

const showLatest = ({latestData}) => {
    if(latestData){
        return latestData.map((data) => {
            return(
                <Link key={data.id} to={`/details/${data.id}`} className="item">
                    <div className="image_cover"
                        style={{background:`url(/images/articles/${data.img})`}}>
                        <div className="description">
                            <span>{data.category}</span>
                            <div>{data.title}</div>
                        </div>
                    </div>
                </Link>
            )
        })
    }
}

const LatestNews = (props) => {
    return(
        <div className="home-latest">
            {showLatest(props)}
        </div>
    )
}

export default LatestNews;
```

Step 24: Finally, start your application using ***npm start.*** In the browser we can see the output where data is received from the API.


localhost:3000
REDUX NEWS
SPORTS
Sed do eiusmod tempor incididunt ut labore et dolore
POLITICS
Similique sunt in culpa qui officia deserunt mollitia animi
SHOWS
Duis aute irure dolor in reprehenderit in voluptate
localhost:3000/details/4
©EDUREKA

Conclusion:

We have successfully created a News Application using React and Redux, where data is recived via an API.