**DAY 27 – USING FETCH API TO POPULATE NEWSITEMS**

Async componentDidMount() {

Let url = 'https://newsapi.org/v2/everything?q=tesla&from=2022-03-17&sortBy=publishedAt&apiKey=52f2b93ef9764c26a72e94eab22c4790'

Let data = await fetch(url)

Let parsedData = await data.json()

Console.log(parsedData)

This.setState({ articles : parsedData.articles })

**DAY 28 – ADDING PREVIOUS AND NEXT BUTTONS**

this.state = {

articles: [],

loading: false,

**page: 1,**

}

DIDMOUNT : this.setState({

articles: parsedData.articles,

**totalResults: parsedData.totalResults,**

})

handleNextClick = async () => {

console.log('Next')

if (this.state.page + 1 > Math.ceil(this.state.totalResults / 20)) {

} else {

let url = `https://newsapi.org/v2/everything?domains=wsj.com&apiKey=52f2b93ef9764c26a72e94eab22c4790&page=${

this.state.page + 1

}&pageSize=20`

let data = await fetch(url)

let parsedData = await data.json()

cons

**DAY 31 – Fetching news category wise**

static defaultProps = {

country: 'in',

pageSize: 2,

category: 'general',

}

static propTypes = {

country: PropTypes.string,

pageSize: PropTypes.number,

category: PropTypes.string,

}

let url = `https://newsapi.org/v2/top-headlines?country=${this.props.country}&category=${this.props.category}&apiKey=52f2b93ef9764c26a72e94eab22c4790&pageSize=${this.props.pageSize}`

<Router>

<Navbar />

<Routes>

<Route

exact

path='/techonology'

element={

<News

pageSize={15}

key='general'

country='in'

category= techonology

/>

}

></Route>

</Routes>

**DAY 32 – Adding time , author and news source**

<span

className='position-absolute top-0 translate-middle badge rounded-pill bg-danger'

style={{ left: '90%', zIndex: 1 }}

>

{source}

</span>

<p className='card-text'>

<small className='text-muted'>

By {!author ? 'Unknown' : author} on{' '}

{new Date(date).toGMTString()}

</small>

</p>

**DAY 33 – Refactoring news component to use the same function**

async updateNews(pageNo)

{

const url = `https://newsapi.org/v2/top-headlines?country=${this.props.country}&category=${this.props.category}&apiKey=52f2b93ef9764c26a72e94eab22c4790&page=${this.state.page}&pageSize=${this.props.pageSize}`

this.setState({ loading: true })

let data = await fetch(url)

let parsedData = await data.json()

this.setState({

articles: parsed Data.articles, totalResults: parsedData.totalResults, loading: false, }) }

**DAY 34 – React component LifeCycle & LifeCycle Methods**

constructor(props) {

super(props)

console.log('Hello I am a constructor from News component')

this.state = {

articles: [],

loading: false,

page: 1,

}

**document.title = `${this.capitalizeFirstLetter(this.props.category)} - NewsMonkey`**

}

<h1 className='text-center'>NewsMonkey - Top {this.capitalizeFirstLetter(this.props.category)} Headlines</h1>

**36. Adding a top loading bar**

import LoadingBar from 'react-top-loading-bar'

<LoadingBar

color='#f11946'

progress={this.state.progress}

/>

state = {

progress:0

}

setProgress=(progress)=>

{

this.setState({progress:progress})

}

<News setProgress={this.setProgress}