SE3040 – Application Frameworks

2024 – Assignment02

BSc (Hons) in Information Technology specializing in Software Engineering

Development of a React Frontend Application Using NASA APIs

Fernando M.A.C

IT21197000

About.js

APOD- Astronomy Picture of the Day

API: https://api.nasa.gov/planetary/apod?api_key=DEMO_KEY

(Display an image by pressing the button)

```
const handleButton = async () => {
  try {
    const res = await axios.get(
      'https://api.nasa.gov/planetary/apod?api_key=DEMO_KEY'
    );
    setImg(res.data);
    console.log(res.data);
  } catch (error) {
    console.error('Error fetching image:', error);
  }
};
```

CMEAnalisys.js

Coronal Mass Ejection

API: https://api.nasa.gov/DONKI/CME?startDate=2017-01-03&endDate=2017-01-03&api_key=DEMO_KEY

(Retrieve the Coronal Mass Ejection)

```
useEffect(() => {
  const fetchData = async () => {
    try {
      const startDate = '2024-03-28'; // Replace with actual start date
      const endDate = '2024-03-29'; // Replace with actual end date
      const apiKey = 'DEMO_KEY'; // Replace with your actual NASA API key

      const response = await fetch('https://api.nasa.gov/DONKI/CME?startDate=2017-01-03&endDate=2017-01-03&api_key=DEMO_KEY');

      if (!response.ok) {
        throw new Error('Failed to fetch data');
      }

      const responseData = await response.json();
      console.log('Data from API:', responseData); // Log the data to see what's being returned
      if (responseData && responseData.length > 0) {
        setData(responseData);
      } else {
        throw new Error('No CME analysis data available for the specified date range');
      }
      setLoading(false);
    } catch (error) {
      console.error('Error fetching data:', error); // Log any errors that occur
      setError(error.message);
      setLoading(false);
    }
  };

  fetchData();
}, []);
```

Details.js

Image of the day

API: https://api.nasa.gov/planetary/apod

(Display the Image of the day that was captured by the NASA)

```
useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await axios.get('https://api.nasa.gov/planetary/apod', {
          params: {
            api_key: 'DEMO_KEY'
          }
        });
        setDetails(response.data);
      } catch (error) {
        console.error('Error fetching APOD data:', error);
      }
    };

    fetchData();
  }, []);
```

EpicImage.js

Epic

API:

1. https://api.nasa.gov/EPIC/api/natural/date/${date}?api_key=${apiKey}
   (Connect with the network)
2. https://epic.gsfc.nasa.gov/archive/natural/${year}/${month}/${day}/jpg/${data[0].image}.jpg
   (Display the image according to the provided data)

(API provides information on the daily imagery collected by DSCOVR's Earth Polychromatic Imaging Camera (EPIC) instrument.)

```
const handleSubmit = async (event) => {
  event.preventDefault();
  try {
    const response = await fetch(`https://api.nasa.gov/EPIC/api/natural/date/${date}?api_key=${apiKey}`);
    if (!response.ok) {
      throw new Error('Network response was not ok');
    }
    const data = await response.json();
    if (Array.isArray(data) && data.length > 0 && typeof data[0].image === 'string') {
      const [year, month, day] = date.split("-");
      setImageStatus('Found');
      setImageSrc(`https://epic.gsfc.nasa.gov/archive/natural/${year}/${month}/${day}/jpg/${data[0].image}.jpg`);
      setImageCaption(data[0].caption);
    } else {
      setImageStatus('Image not found for the selected date');
      setImageSrc('');
      setImageCaption('');
    }
  } catch (error) {
    console.error('Error in network request:', error);
    setImageStatus('Please try a different date or check your syntax!');
  }
};
```

NASAImage.js

Earth

API: https://api.nasa.gov/planetary/earth/imagery?lon=-95.33&lat=29.78&date=2018-01-01&dim=0.15&api_key=DEMO_KEY

(Display an image of the Earth)

```
useEffect(() => {
  const fetchData = async () => {
    try {
      const response = await fetch(
        'https://api.nasa.gov/planetary/earth/imagery?lon=-95.33&lat=29.78&date=2018-01-01&dim=0.15&api_key=DEMO_KEY'
      );
      // Check if the response is successful
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      // If the response is an image, get the blob URL
      const blob = await response.blob();
      setImageUrl(URL.createObjectURL(blob));
    } catch (error) {
      console.error('Error fetching data from NASA API:', error);
    }
  };

  fetchData();
}, []);
```

NEOData.js

NEO

API:
https://api.nasa.gov/neo/rest/v1/feed?start_date=${startDate}&end_date=${endDate}&api_key=${apiKey}

(RESTful web service for near earth Asteroid information)

```
const fetchData = () => {
  const apiKey = 'DEMO_KEY';
  const url = `https://api.nasa.gov/neo/rest/v1/feed?start_date=${startDate}&end_date=${endDate}&api_key=${apiKey}`;

  axios.get(url)
    .then((response) => {
      const data = response.data;
      const neoObj = data.near_earth_objects;
      const dates = Object.keys(neoObj);
      const neoData = dates.map((date) => neoObj[date]);
      setNeoData(neoData);
    })
    .catch((error) => {
      console.error('There was a problem with the fetch operation:', error);
    });
};
```