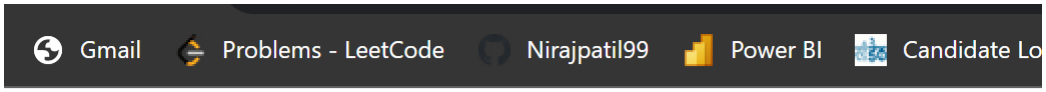
Assignment 14

TVM-DAC-Niraj Patil

Q1. Write a react js program to display the greeting ‘Hello {name}, Welcome to {place}’ using React.Fragment.(Hint: Hello Abhijith, Welcome to Chembur).

```
import React from 'react'

function Greeting() {
    const name = 'Abhijith';
  const place = 'Chembur';
  return (
    <React.Fragment>
      <h1>Hello {name},</h1>
      <p>Welcome to {place}</p>
    </React.Fragment>
  )
}

export default Greeting
```

output:



Hello Abhijith,

Welcome to Chembur

Q2. Write a React js program to display elements in a table.

Name Address

Dhoni Ranchi

Kohli Delhi

Jadeja Gujarat

```
import React from 'react';
function Table(props) {
  return (
    <table>
      <thead>
        <tr>
          <th>Name</th>
         <th>Address</th>
        </tr>
      </thead>
      <tbody>
        {props.data.map((item, index) => (
          <tr key={index}>
            <td>{item.name}</td>
            <td>{item.Address}</td>
          </tr>
        ))}
      </tbody>
    </table>
  );
}

export default Table;
```

```
import logo from './logo.svg';
import './App.css';
import React,{useState} from 'react';
import Welcome from './componants/Welcome';
import ClassDemo from './componants/ClassDemo';
import Greeting from './componants/Greeting';
import Table from './componants/Table';

function App() {
  const data = [
    { name: 'Dhoni', Address:'Ranchi' },
    { name: 'kolhi', Address:'Delhi' },
    { name: 'jadeja', Address:'Gujarat' },
  ];
  return (
    <div className="App">
    <h1>Table Example</h1>
```

```
    <Table data={data} />
  </div>


  );
}

export default App;
```

output:

Name	Address
Dhoni	Ranchi
kolhi	Delhi
jadeja	Gujarat

Q3. Create a react application to display the message “I am Santhosh. I teach at Indore.” using the props keyword.

```
import React from 'react';

function TeacherInfo(props) {
  return (
    <div>
      <p>{props.message}</p>
    </div>
  );
}

export default TeacherInfo;
```

```
import logo from './logo.svg';
import React from 'react';
import './App.css';
import TeacherInfo from './componants/TecherInfo';
function App() {
```

```
  return (
    <div className="App">
    <TeacherInfo message="I am Santhosh. I teach at Indore." />
  </div>
  );
}

export default App;
```

output:

I am Santhosh. I teach at Indore.

Q4. Write a React js program to demonstrate how data can be modified using a class component. On the click of a button, the message ‘Thank you for using my React application.’ must be displayed.

```
import React, { Component } from 'react';

class MessageModifier extends Component {
  constructor(props) {
    super(props);
    this.state = {
      message: 'Click the button to modify the message.',
    };
  }

  modifyMessage = () => {
    this.setState({
      message: 'Thank you for using my React application.',
    });
  };

  render() {
    return (
      <div>
        <h1>{this.state.message}</h1>
        <button onClick={this.modifyMessage}>Modify Message</button>
      </div>
    );
  }
}
```

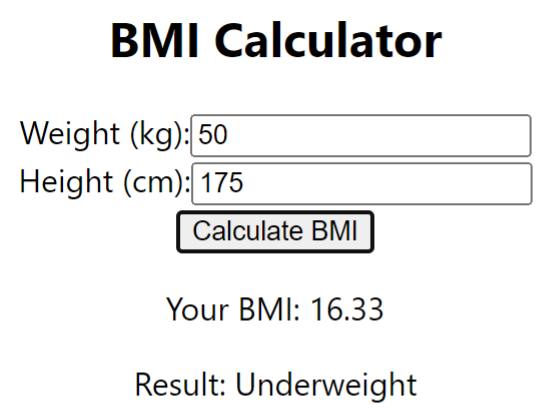
```
export default MessageModifier;
```

Q5. Design a React application to calculate the BMI of human beings. If the BMI<18, the output message should be displayed as ‘underweight’,if BMI is between 18 and 25, the output message should be displayed as ‘Normal’, if weight is between 25 and 39, the output must be displayed as ‘overweight’ and if bmi>39, the output must be displayed as ‘overweight.’ Handle the exception of alphanumeric inputs.

```
import React, { useState } from 'react';

const BMICalculator = () => {
  const [weight, setWeight] = useState('');
  const [height, setHeight] = useState('');
  const [bmi, setBMI] = useState(null);
  const [result, setResult] = useState('');

  const calculateBMI = () => {
    try {
      const weightValue = parseFloat(weight);
      const heightValue = parseFloat(height);

      if (isNaN(weightValue) || isNaN(heightValue)) {
```

```
        throw new Error('Please enter valid numeric values for weight and
height.');
      }

      const bmiValue = weightValue / ((heightValue / 100) * (heightValue /
100));
      setBMI(bmiValue.toFixed(2));

      if (bmiValue < 18) {
        setResult('Underweight');
      } else if (bmiValue >= 18 && bmiValue <= 25) {
        setResult('Normal');
      } else if (bmiValue > 25 && bmiValue <= 39) {
        setResult('Overweight');
      } else {
        setResult('Obese');
      }
    } catch (error) {
      setResult('Error: ' + error.message);
    }
  };

  return (
    <div>
      <h2>BMI Calculator</h2>
      <div>
        <label>Weight (kg):</label>
        <input type="text" value={weight} onChange={(e) =>
setWeight(e.target.value)} />
      </div>
      <div>
        <label>Height (cm):</label>
        <input type="text" value={height} onChange={(e) =>
setHeight(e.target.value)} />
      </div>
      <button onClick={calculateBMI}>Calculate BMI</button>
      {bmi !== null && (
        <div>
          <p>Your BMI: {bmi}</p>
          <p>Result: {result}</p>
        </div>
      )}
    </div>
  );
};

export default BMICalculator;
```

output:



Q6.

```
import React, { useState } from 'react';
import ScoreDisplay from './ScoreDisplay';
import ScoreButtons from './ScoreButtons';

const ScoreCalculator = () => {
  const [score, setScore] = useState(0);

  const incrementScore = () => {
    setScore(score + 1);
  };

  const decrementScore = () => {
    setScore(score - 1);
  };

  return (
    <div>
      <h2>Score Calculator App</h2>
      <ScoreDisplay score={score} />
      <ScoreButtons onIncrement={incrementScore} onDecrement={decrementScore}
/>
    </div>
  );
};
```

```
export default ScoreCalculator;
import React from 'react';

const ScoreDisplay = ({ score }) => {
  return (
    <div>
      <p>Current Score: {score}</p>
    </div>
  );
};

export default ScoreDisplay;
```

```
import React from 'react';

const ScoreButtons = ({ onIncrement, onDecrement }) => {
  return (
    <div>
      <button onClick={onIncrement}>Increment Score</button>
      <button onClick={onDecrement}>Decrement Score</button>
    </div>
  );
};

export default ScoreButtons;
```

Output: