**Assignment 15**

**TVM-DAC-Niraj Patil**

Q1. Design a React application to calculate the time saved by an employee. The application

must look like this:

Saved Time= {Punch out time -Punch in Time} - 8 hours. Display the saved time in

minutes. Handle the necessary exceptions that could occur.

import React, { useState } from 'react';

const TimeCalculator = () => {

  const [punchInTime, setPunchInTime] = useState('');

  const [punchOutTime, setPunchOutTime] = useState('');

  const [savedTime, setSavedTime] = useState(null);

  const [error, setError] = useState('');

  const calculateSavedTime = () => {

    try {

      const punchIn = new Date(`2023-01-01T${punchInTime}`);

      const punchOut = new Date(`2023-01-01T${punchOutTime}`);

      if (isNaN(punchIn) || isNaN(punchOut)) {

        throw new Error('Invalid time format. Please use HH:mm.');

      }

      const timeDifference = punchOut - punchIn - 8 \* 60 \* 60 \* 1000; // Subtract 8 hours in milliseconds

      const savedMinutes = Math.floor(timeDifference / (60 \* 1000));

      setSavedTime(savedMinutes);

      setError('');

    } catch (err) {

      setError(err.message);

      setSavedTime(null);

    }

  };

  return (

    <div>

      <h2>Time Saver Calculator Form</h2>

      <div>

        <label>Punch In Time:</label>

        <input

          type="time"

          value={punchInTime}

          onChange={(e) => setPunchInTime(e.target.value)}

        />

      </div><br></br>

      <div>

        <label>Punch Out Time:</label>

        <input

          type="time"

          value={punchOutTime}

          onChange={(e) => setPunchOutTime(e.target.value)}

        />

      </div> <br></br>

      <button onClick={calculateSavedTime}>Calculate Saved Time</button>

      {savedTime !== null && <p>Saved Time: {savedTime} minutes</p>}

      {error && <p style={{ color: 'red' }}>{error}</p>}

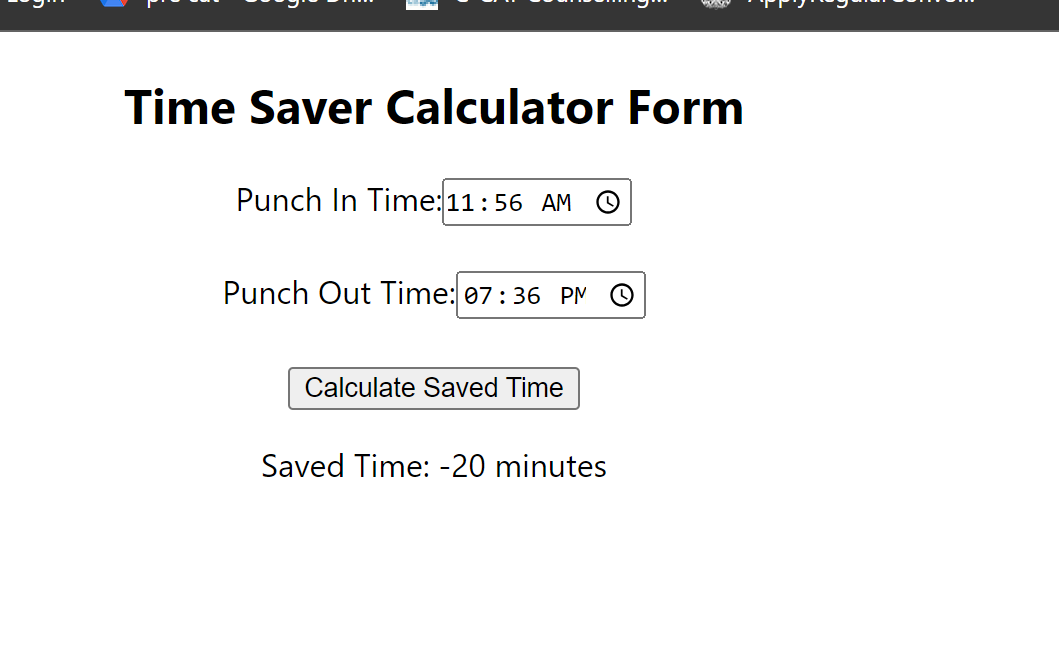
    </div>

  );

};

export default TimeCalculator;

output:



Q2. Create a react application to display a paragraph of text under which there are two buttons like and dislike.When the user clicks the button, the number of likes and dislikes should be shown near the button. When the user clicks the button, increment the respective counter.

import React,{useState} from 'react';

const LikeDislike=()=>{

    const [like,setLike]=useState(0);

    const [dislike,setDisLike]= useState(0);

    const handleLike=()=>{

        setLike(like+1);

    }

    const handleDislike=()=>{

        setDisLike(dislike+1);

    }

    return(

        <div>

            <p>

            React makes it painless to create interactive UIs.Design simple views for each state in your application,

            and React will efficiently update and render just the right components when your data changes.

            we can use javascript along with html in react.

            we also use bootstrap for making dynamic and interactive web application.

            </p>

            <button onClick={handleLike}>Like ({like})</button> <br></br>

            <button onClick={handleDislike}>Dislike ({dislike})</button>

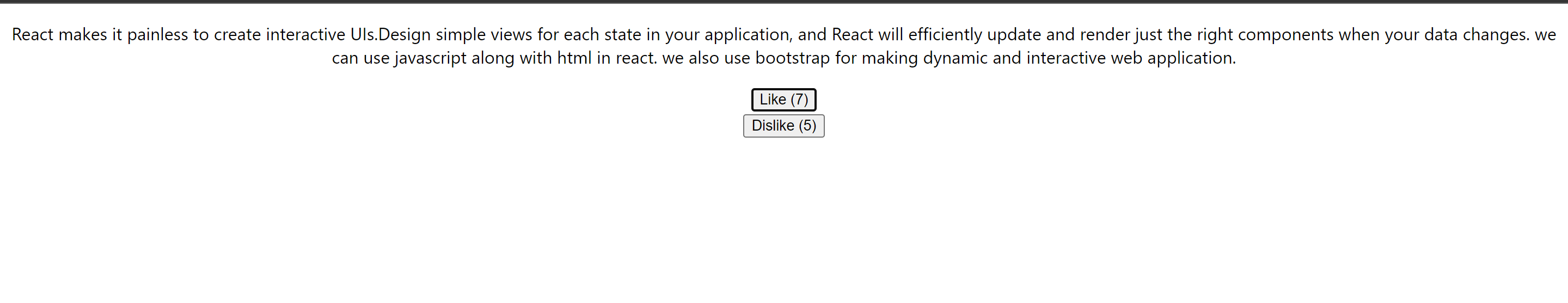
        </div>

    )

}

export default LikeDislike;

output:



Q3. Write a react program to display an array of marks , Display the details of each student such as Id, Name etc.

import React from 'react'

function StudentDetail() {

    const studentmarks=[

        {id:1, name:'Niraj', marks:75},

        {id:2, name:'Yash', marks:67},

        {id:3, name:'tejas', marks:85},

        {id:4, name:'Sorubh', marks:87},

    ]

  return (

    <div>

        <h2>StudentDetail</h2>

        <ul>

        {studentmarks.map((student) => (

          <li key={student.id}>

            <h3>ID:</h3> {student.id}

            <h3>Name:</h3> {student.name}

            <h3>Subject:</h3> {student.subject}

            <h3>Marks:</h3> {student.marks}

            <hr />

          </li>

        ))}

      </ul>

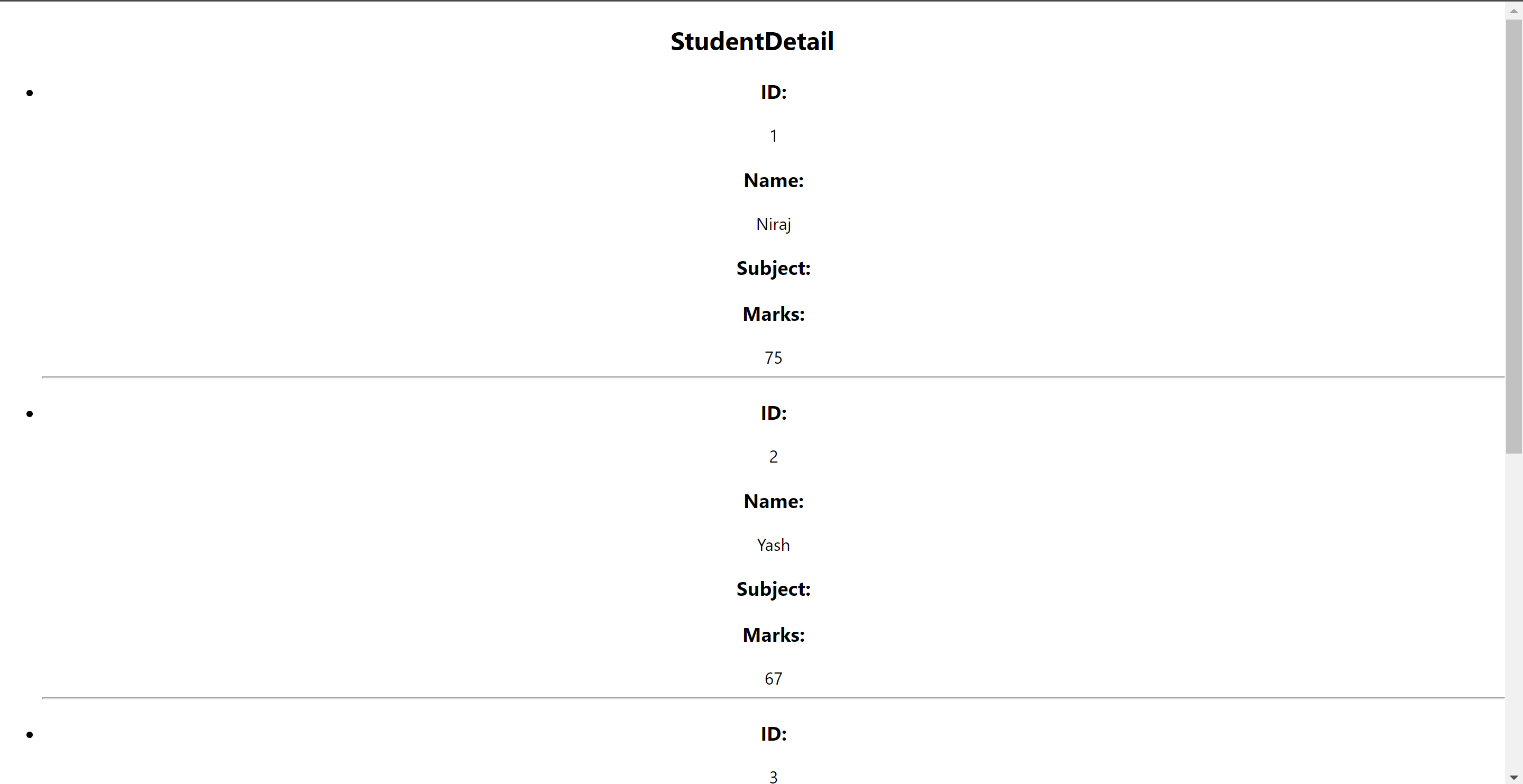
    </div>

  )

}

export default StudentDetail

output:



Q4. Using react, demonstrate how array can be disabled in a component.

import React, { useState } from 'react';

const DisableArrayComponents = () => {

  const [isDisabled, setIsDisabled] = useState(true);

  const handleChangeDisable = () => {

    setIsDisabled(!isDisabled);

  };

  const inputArray = [

    { id: 1, label: 'Input 1' },

    { id: 2, label: 'Input 2' },

    { id: 3, label: 'Input 3' },

  ];

  return (

    <div>

      <h2>Disabled Array Component</h2>

      <button onClick={handleChangeDisable}>

        {isDisabled ? 'Enable' : 'Disable'} Inputs

      </button>

      <form>

        {inputArray.map((input) => (

          <div key={input.id}>

            <label>{input.label}</label>

            <input

              type="text"

              placeholder={`Enter ${input.label}`}

              disabled={isDisabled}

            />

          </div>

        ))}

      </form>

    </div>

  );

};

export default DisableArrayComponents;

output:

