**2021-08-14**

**Counter down JAVASCRIPT code for PARIKHSA page.**

**<CODE>**

duration="90"; // time should be in minutes

let hr = (Math.floor(duration/60));

let min= duration%60;

// let dur=new String(hr+":"+min+":00");

let [remaing\_time,setTime]=useState(hr+":"+min+":00");

(()=>{

    let time=remaing\_time.split(":");

    let  h=time[0];

    let  m=time[1];

    let  s=time[2];

  let  stop\_time=  setInterval(() => {

    if(s>0){ s=s-1;  s =s<=9?"0"+s:s;  }

    else{s=59 ;

     if(m>0){m=m-1; m= m<10?"0"+m:m ; }

     else{m=59;

      if(h>0){h=h-1; h=h<10?"0"+h:h;}

      else{clearInterval(stop\_time); s="00";m="00"; }

         }

       }

    setTime(`${h}:${m}:${s}`);

   },1000);

})()

// End code for Timer

**</CODE>**

**2021-08-10**

**Code for some Graphs**

**Table**

// TableData  for tableChart

  let  tableData=[

    [

      {type: 'string', label: 'EXAM ID' },

      {type: 'number', label: 'No. of Questions' },

      {type: 'number', label: 'Total Marks' },

      {type: 'number', label: 'Attempted'},

      {type: 'number', label: 'Unattempted'},

      {type: 'number', label: 'Correct'},

      {type: 'number', label: 'Incorrect'},

      {type: 'number', label: 'Marks Obtained'}

    ]

  ]

  //  tableData[1]=[123,12,74,540,4240,440,120,22];

  //  tableData[2]=[123,12,74,540,4240,440,120,22];

  //  tableData[3]=[13,12,74,540,4240,440,120,22];

<Chart

  width={'1500px'}

  height={'300px'}

  chartType="Table"

  loader={<div>Loading Chart</div>}

  data={tableData}

  options={{

    showRowNumber: true,

  }}

  rootProps={{ 'data-testid': '1' }}

/>

**CART BAR**

 let  results;

//  results = [

  //   ['Result', 'appeared'],

  //   ['correct', 45],

  //   ['incorrect', 65],

  // ]

<Chart

  width={'500px'}

  height={'300px'}

  chartType="PieChart"

  loader={<div>Loading Chart</div>}

  data={results}

  options={{

      title: 'AIEEE Mock Test performance',

      subtitle: 'attempt wise growth',

    }}

  rootProps={{ 'data-testid': '1' }}

/>

**9/2/2021**

**Code for Side Notices**

// Right side board

const  RightBoard  =(props)=>{

      let   rstyle= { // RightBoardStyle

        position:"fixed",

        top:"16%",

        right:"2%"

    };

     let mq= Array.from(new Set(props.mQ)); //Array of unique ele

     let aq= Array.from(new Set(props.aQ));

    return (<>

             <table style={rstyle} >

              <tr>

  <td> <span class="badge badge-danger p-2 mr-1 ml-1 "> Mark for review </span>  </td>

    <td><span class="badge badge-success p-2  mr-1 ml-1 ">Answered</span>  </td>

            </tr>

               <tr>

          {/\* Mark for review  \*/}

                 <td>

                 <div style={{maxWidth:"100px"}} >

              { mq.map(e=>{

                return  <div style={{width:"25px",cursor:"pointer",float:"left" }}  onClick={()=>{props.setIndex(e-1)}}

                class="m-1 bg-danger text-center rounded-circle text-white">

                     {e}

                 </div>

             })

             }

              </div>

                 </td>

       {/\* Answered question  \*/}

                 <td>

                 <div style={{maxWidth:"100px"}} >

              { aq.map(e=>{

                return  <div style={{width:"25px",cursor:"pointer",float:"left"}}  onClick={()=>{props.setIndex(e-1)}}

                class="m-1 bg-success text-center rounded-circle text-white"> {e} </div>

             })

             }

              </div>

                 </td>

               </tr>

           </table>

                {/\* </div> \*/}

            </>);

    }

9/2/2021

Code for Question options

  <input name={current.\_id} value='a' className="form-check-input" type="radio" id="a"

               checked={answers[index] ? answers[index].option==='a'? true:false:false}

         onClick ={()=>{

             answers[index] ? answers[index].option==='a'? delete answers[index] : delete answers[index] :

             answers[index]={answer: current.options[0], option: 'a'}

               setAnswers(answers);

              aQ.includes(index+1)?

              setaQ(aQ.filter(e=>e!=index+1))

              :setaQ([...aQ,index+1]);

             }}/>

         <label className="form-check-label" htmlFor="formCheck-1">{current.options[0]}</label>

Code for Cursor

 let cur1={

        cursor:"pointer"

    }

    let cur2={

        cursor:"not-allowed"

    }

Code for changing style depending on index values

  style={index===0?cur2:cur1}

Code for changing index

<button id="prev-btn" disabled={index === 0 ? true : false }

                style={index===0?cur2:cur1}

               className="btn btn-outline-secondary" type="button" onClick={()=>{setIndex(index-1) }}>

              Previous</button>

Code for marked question and removed from marked

 <div className="ml-auto"><button className="btn btn-info" type="button"

             onClick={()=>{

              mQ.includes(index+1)?

              setmQ(mQ.filter(e=>e!==index+1))

             :setmQ([index+1,...mQ])

            }

            }

             style={cur1}

            >{mQ.includes(index+1)?"Removed from marked":" Mark for Review"}</button>