|  |  |  |
| --- | --- | --- |
| **Name: ABHIN G DAS** | **SRN: PES2UG23CS019** | **Section: A** |
| **Date: 04-11-2024** | **Unit 3 Assignment Exercise** |
| **PROBLEM STATEMENT-1 (for odd number SRNs):** | | |
| Create and display a list of 3 patient appointments using the React framework. Each appointment should include the patient's name, reason for appointment, appointment date, and status (e.g., scheduled, completed). When a user clicks a button, the status of the selected appointment should change to "completed", and the total number of completed appointments should be recalculated and displayed. Use a style object variable for custom styling, like how you would in JavaScript. Ensure that the appointment details are read from an array of objects | | |
| **OBJECTIVE** | | |
| The objective of this exercise is to test the student on the front-end framework React JS. It evaluates the student’s knowledge React Js components, ReactJS props and state, ReactJS Lists and Keys, React JS events | | |
| **PREREQUISITE** | | |
| In order to complete this exercise, the student needs to understand the fundamentals of HTML, CSS, and JavaScript | | |

|  |
| --- |
| **SAMPLE1:**  Here's a visual representation of what the webpage will look like:  Explanation:  **Sample UI Design Picture**  **Patient Appointments**  **Description:**   1. **Header**: "Patient Appointments" is prominently displayed at the top, centered. 2. **Appointment Cards**:    * Each card includes:      + **Patient Name**: Large, bold text.      + **Reason**: Description of the appointment reason.      + **Date**: The date of the appointment.      + **Status**: Current status of the appointment (Scheduled/Completed).      + **Button**: A button to mark the appointment as completed, only visible if the appointment is scheduled.    * Completed appointments have the name crossed out and in gray color. 3. **Total Completed Appointments**: Displays the total number of completed appointments at the bottom, centered and bold.   **SAMPLE SCREENSHOT OF OUTPUT (Just for your reference)** |
| **PROGRAM** |
| AppointmentList.js  import React, { useState } from 'react';  const initialAppointments = [    { id: 1, name: 'Rahul Kumar', reason: 'Routine Checkup', date: '2024-11-01', status: 'scheduled' },    { id: 2, name: 'Rohit Urus', reason: 'Dental Cleaning', date: '2024-11-03', status: 'scheduled' },    { id: 3, name: 'Kapil Dev', reason: 'Full Body Check-up', date: '2024-11-05', status: 'Appointment yet-to-take' }  ];  const AppointmentList = () => {    const [appointments, setAppointments] = useState(initialAppointments); //usestate is hook to manage state in functional component    // Count completed appointments    const completedCount = appointments.filter(*app* => app.status === 'completed').length; //filter is used to filter the array based on condition    // This is to make  the appointments list to be updated    const markAsCompleted = (*id*) => {      setAppointments(*prevAppointments* =>        prevAppointments.map(*app* =>          app.id === id ? { ...app, status: 'completed' } : app        )      ); //map is used to iterate over the array and update the array based on condition      //markascompleted is a function which is called when the button is clicked and it updates the status of the appointment to completed    }; //setAppointments is used to update the state of the appointments    const styles = {      container: {        maxWidth: '800px',        margin: '20px auto',        padding: '30px',        backgroundColor: '#f0f4f8',        borderRadius: '15px',        boxShadow: '0 10px 20px rgba(0, 0, 0, 0.1)'      },      header: {        fontSize: '32px',        marginBottom: '30px',        color: '#2c3e50',        textAlign: 'center',        fontWeight: 'bold'      },      card: {        padding: '25px',        border: 'none',        borderRadius: '12px',        marginBottom: '20px',        backgroundColor: '#fff',        boxShadow: '0 5px 15px rgba(0, 0, 0, 0.08)',        transition: 'all 0.3s ease'      },      cardHover: {        transform: 'translateY(-5px)',        boxShadow: '0 8px 20px rgba(0, 0, 0, 0.12)'      },      completedCard: {        backgroundColor: '#e8f5e9',        opacity: 0.8      },      patientName: {        fontSize: '24px',        color: '#34495e',        marginBottom: '10px'      },      appointmentInfo: {        fontSize: '16px',        color: '#7f8c8d',        marginBottom: '5px'      },      statusTag: {        display: 'inline-block',        padding: '5px 10px',        borderRadius: '20px',        fontSize: '14px',        fontWeight: 'bold',        textTransform: 'uppercase'      },      scheduledStatus: {        backgroundColor: '#3498db',        color: '#fff'      },      completedText: {        textDecoration: 'line-through',        color: 'gray'      },        completedStatus: {        backgroundColor: '#2ecc71',        color: '#fff'      },      button: {        marginTop: '15px',        padding: '10px 20px',        cursor: 'pointer',        backgroundColor: '#3498db',        color: '#fff',        border: 'none',        borderRadius: '25px',        fontSize: '16px',        transition: 'all 0.3s ease',        outline: 'none'      },      buttonHover: {        backgroundColor: '#2980b9',        transform: 'scale(1.05)'      },      footer: {        marginTop: '30px',        fontSize: '18px',        fontWeight: 'bold',        color: '#2c3e50',        textAlign: 'center'      }    };    return (      <div style={styles.container}>        <h1 style={styles.header}>Patient Appointments</h1>          {appointments.map(*app* => (          <div            key={app.id}            style={{ ...styles.card, ...(app.status === 'scheduled' ? styles.cardHover : {}) }}          >            <h2 style={app.status === 'completed' ? styles.completedText : {}}>{app.name}</h2>            <p><strong>Reason:</strong> {app.reason}</p>            <p><strong>Date:</strong> {app.date}</p>            <p><strong>Status:</strong> {app.status}</p>            {app.status === 'scheduled' && (              <button                style={styles.button}                onMouseOver={(*e*) => e.currentTarget.style.backgroundColor = '#45a049'}                onMouseOut={(*e*) => e.currentTarget.style.backgroundColor = '#4CAF50'}                onClick={() => markAsCompleted(app.id)}              >                Mark as Completed              </button>            )}          </div>        ))}        <div style={styles.footer}>          Total Completed Appointments: {completedCount}        </div>      </div>    );  };  export default AppointmentList;  app.css  .App {    text-align: center;  }  .App-logo {    height: 40vmin;    pointer-events: none;  }  @media (prefers-reduced-motion: no-preference) {    .App-logo {      animation: App-logo-spin infinite 20s linear;    }  }  .App-header {    background-color: #282c34;    min-height: 100vh;    display: flex;    flex-direction: column;    align-items: center;    justify-content: center;    font-size: calc(10px + 2vmin);    color: white;  }  .App-link {    color: #61dafb;  }  @keyframes *App-logo-spin* {    from {      transform: rotate(0deg);    }    to {      transform: rotate(360deg);    }  } |
| **SCREENSHOT OF YOUR OUTPUT** |
|  |