**Objective:-** Create an application for student result management using React JS.

**Code :-**

**App.jsx**

import React, { useState } from 'react';

import AddMarks from './components/AddMarks';

import MarksTable from './components/MarksTable';

import './App.css';  *// Styling import*

const App = () => {

  const [marks, setMarks] = useState([]);

  const [editingMark, setEditingMark] = useState(**null**);

*// Add or update student marks*

  const addMark = (*newMark*) => {

    const total = Object.values(newMark.subjects).reduce(

      (*acc*, *val*) => acc + parseInt(val || 0),

      0

    );

    const percentage = ((total / 600) \* 100).toFixed(2);

    if (editingMark) {

*// Update the existing mark*

      const updatedMarks = marks.map((*mark*) =>

        mark.id === editingMark.id

          ? { ...newMark, id: editingMark.id, total, percentage }

          : mark

      );

      setMarks(updatedMarks);

      setEditingMark(**null**);  *// Clear edit mode*

    } else {

*// Add new mark*

      setMarks([

        ...marks,

        { ...newMark, id: Date.now(), total, percentage }

      ]);

    }

  };

  const handleEdit = (*mark*) => {

    setEditingMark(mark);

  };

  const deleteMark = (*id*) => {

    setMarks(marks.filter((*mark*) => mark.id !== id));

    setEditingMark(**null**);  *// Exit edit mode on delete*

  };

  return (

    <div className="container">

      <h1>Student Result Management</h1>

      <AddMarks onAdd={addMark} editingMark={editingMark} />

      <MarksTable marks={marks} onEdit={handleEdit} onDelete={deleteMark} />

    </div>

  );

};

export default App;

**App.css**

*/\* App.css \*/*

*/\* Import Poppins Font \*/*

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap');

*/\* General Reset \*/*

\* {

  box-sizing: border-box;

  margin: 0;

  padding: 0;

}

*/\* Full-Screen Layout \*/*

body {

  font-family: 'Poppins', sans-serif;

  background: linear-gradient(135deg, #74EBD5, #ACB6E5);

  color: #333;

  min-height: 100vh;

  display: flex;

  justify-content: center;

  align-items: center;

  padding: 20px;

}

*/\* Main Container \*/*

.container {

  background: #fff;

  box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);

  border-radius: 12px;

  max-width: 1200px;

  width: 100%;

  padding: 40px;

  animation: fadeIn 0.8s ease-in-out;

  transition: all 0.3s;

}

*/\* Header Styling \*/*

h1 {

  color: #4CAF50;

  font-size: 36px;

  margin-bottom: 20px;

  text-align: center;

}

*/\* Form Styling \*/*

form {

  display: grid;

  grid-template-columns: repeat(4, 1fr);

  gap: 20px;

  padding: 20px 0;

}

form input {

  padding: 12px;

  border: 1px solid #ccc;

  border-radius: 8px;

  font-size: 16px;

  color: #333;

  transition: all 0.3s;

}

form input*:focus* {

  outline: none;

  border-color: #4CAF50;

  box-shadow: 0 0 8px rgba(76, 175, 80, 0.5);

}

form button {

  grid-column: span 4;

  padding: 14px;

  background: #4CAF50;

  color: white;

  border: none;

  border-radius: 8px;

  font-size: 16px;

  cursor: pointer;

  transition: background 0.3s, transform 0.2s;

}

form button*:hover* {

  background: #45a049;

  transform: scale(1.05);

}

form button*:active* {

  transform: scale(0.95);

}

*/\* Table Styling \*/*

table {

  width: 100%;

  border-collapse: collapse;

  margin-top: 30px;

}

thead {

  background: #4CAF50;

  color: white;

}

th, td {

  padding: 15px;

  text-align: center;

  border-bottom: 1px solid #ddd;

  font-size: 16px;

}

tbody tr*:hover* {

  background: #f1f1f1;

}

*/\* Alternating row colors \*/*

tbody tr*:nth-child*(even) {

  background: #f9f9f9;

}

*/\* Button Styling \*/*

button {

  cursor: pointer;

  transition: all 0.3s ease;

}

*/\* Edit and Delete Buttons \*/*

.edit-btn {

  background: #2196F3;

  color: white;

  border: none;

  padding: 10px 20px;

  border-radius: 8px;

  font-size: 14px;

  margin-right: 5px;

  transition: 0.3s;

  margin-bottom: 2px;

  width: 90px;

}

.edit-btn*:hover* {

  background: #1976D2;

  transform: scale(1.05);

}

.delete-btn {

  background: #f44336;

  color: white;

  border: none;

  padding: 10px 20px;

  border-radius: 8px;

  font-size: 14px;

  transition: 0.3s;

  width: 90px;

}

.delete-btn*:hover* {

  background: #d32f2f;

  transform: scale(1.05);

}

*/\* Animation \*/*

@keyframes *fadeIn* {

  from {

    opacity: 0;

    transform: translateY(-30px);

  }

  to {

    opacity: 1;

    transform: translateY(0);

  }

}

*/\* Responsive Design \*/*

@media (max-width: 1024px) {

  form {

    grid-template-columns: repeat(2, 1fr);

  }

  form button {

    grid-column: span 2;

  }

  th, td {

    font-size: 14px;

    padding: 12px;

  }

}

@media (max-width: 768px) {

  form {

    grid-template-columns: 1fr;

  }

  form button {

    grid-column: span 1;

  }

  th, td {

    font-size: 12px;

    padding: 10px;

  }

  .container {

    padding: 20px;

  }

  h1 {

    font-size: 28px;

  }

}

**Components/AddMarks.jsx**

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

const AddMarks = ({ *onAdd*, *editingMark* }) => {

  const [rollNumber, setRollNumber] = useState('');

  const [name, setName] = useState('');

  const [subjects, setSubjects] = useState({

    math: '',

    science: '',

    english: '',

    history: '',

    geography: '',

    computer: ''

  });

*// Pre-fill form with existing data during editing*

  useEffect(() => {

    if (editingMark) {

      setRollNumber(editingMark.rollNumber);

      setName(editingMark.name);

      setSubjects({ ...editingMark.subjects });

    } else {

      setRollNumber('');

      setName('');

      setSubjects({

        math: '',

        science: '',

        english: '',

        history: '',

        geography: '',

        computer: ''

      });

    }

  }, [editingMark]);

  const handleSubmit = (*e*) => {

    e.preventDefault();

    if (rollNumber && name && Object.values(subjects).every((*mark*) => mark !== '')) {

      onAdd({ rollNumber, name, subjects });

*// Clear form after submission*

      setRollNumber('');

      setName('');

      setSubjects({

        math: '',

        science: '',

        english: '',

        history: '',

        geography: '',

        computer: ''

      });

    }

  };

  const handleChange = (*e*) => {

    setSubjects({ ...subjects, [e.target.name]: e.target.value });

  };

  return (

    <form onSubmit={handleSubmit}>

        <input

        type="text"

        placeholder="Student Name"

        value={name}

        onChange={(*e*) => setName(e.target.value)}

        required

      />

      <input

        type="text"

        placeholder="Roll Number"

        value={rollNumber}

        onChange={(*e*) => setRollNumber(e.target.value)}

        required

      />

      {Object.keys(subjects).map((*subject*) => (

        <input

          key={subject}

          name={subject}

          type="number"

          placeholder={`${subject.charAt(0).toUpperCase() + subject.slice(1)} Marks`}

          value={subjects[subject]}

          onChange={handleChange}

          required

        />

      ))}

      <button type="submit">

        {editingMark ? 'Update Marks' : 'Add Marks'}

      </button>

    </form>

  );

};

export default AddMarks;

**Components/MarksTable.jsx**

import React from 'react';

const MarksTable = ({ *marks*, *onEdit*, *onDelete* }) => {

  return (

    <table>

      <thead>

        <tr>

          <th>Roll No</th>

          <th>Name</th>

          <th>Math</th>

          <th>Science</th>

          <th>English</th>

          <th>History</th>

          <th>Geography</th>

          <th>Computer</th>

          <th>Total</th>

          <th>Percentage</th>

          <th>Actions</th>

        </tr>

      </thead>

      <tbody>

        {marks.map((*mark*) => (

          <tr key={mark.id}>

            <td>{mark.rollNumber}</td>

            <td>{mark.name}</td>

            {Object.values(mark.subjects).map((*val*, *idx*) => (

              <td key={idx}>{val}</td>

            ))}

            <td>{mark.total}</td>

            <td>{mark.percentage}%</td>

            <td>

              <button className="edit-btn" onClick={() => onEdit(mark)}>

                Edit </button>

              <button className="delete-btn" onClick={() => onDelete(mark.id)}>

                Delete </button>

            </td>

          </tr>

        ))}

      </tbody>

    </table>

  );

};

export default MarksTable;

**Output :-**





