

FULL STACK WEB DEVELOPMENT

ASSIGNMENT-2

1602-21-733-064

K.Vishal Reddy

CSE-A

1) Develop a web application which keeps track of data about students academic background and helps HR manager to generate eligibility list wrt a Company. There should be provision to capture the data of a candidate through multiple forms. Use the appropriate styling for creating rich interface and perform the necessary validation using JavaScript.

Program:

```
// StudentForm.js

import React, { useState } from 'react';

const StudentForm = ({ onSubmit }) => {
  const [name, setName] = useState('');
  const [rollNumber, setRollNumber] = useState('');
  const [cgpa, setCGPA] = useState('');
  const [backlogs, setBacklogs] = useState('');
  const [branch, setBranch] = useState('');
  const [languages, setLanguages] = useState('');
  const handleSubmit = (e) => {
    e.preventDefault();
    onSubmit({
      name,
      rollNumber,
      cgpa: parseFloat(cgpa),
      backlogs: parseInt(backlogs),
      branch,
      languages: languages.split(',').map((lang) => lang.trim()),
    });
  };
  return (
    <form onSubmit={handleSubmit}>
      <div>
```

FULL STACK WEB DEVELOPMENT

```
        <label>Name:</label>

        <input type="text" value={name} onChange={(e) =>
setName(e.target.value)} required />

    </div>

    <div>

        <label>Roll Number:</label>

        <input type="text" value={rollNumber} onChange={(e) =>
setRollNumber(e.target.value)} required />

    </div>

    <div>

        <label>CGPA:</label>

        <input type="number" step="0.01" value={cgpa} onChange={(e) =>
setCGPA(e.target.value)} required />

    </div>

    <div>

        <label>Backlogs:</label>

        <input type="number" value={backlogs} onChange={(e) =>
setBacklogs(e.target.value)} required />

    </div>

    <div>

        <label>Branch:</label>

        <input type="text" value={branch} onChange={(e) =>
setBranch(e.target.value)} required />

    </div>

    <div>

        <label>Languages Known (comma-separated):</label>

        <input type="text" value={languages} onChange={(e) =>
setLanguages(e.target.value)} required />

    </div>

    <div>

        <button type="submit">Save and next</button>

    </div>

</form>

);

};
```

FULL STACK WEB DEVELOPMENT

```
export default StudentForm;

// StudentList.js

import React from 'react';

const StudentList = ({ students, showAllDetails }) => {
  return (
    <div className="students">
      <h2>All Students</h2>
      <table>
        <thead>
          <tr>
            <th>Name</th>
            <th style={{ whiteSpace: 'nowrap' }}>Roll Number</th>
            <th>CGPA</th>
            <th>Backlogs</th>
            <th>Branch</th>
            <th style={{ whiteSpace: 'nowrap' }}>Languages Known</th>
          </tr>
        </thead>
        <tbody>
          {students.map((student, index) => (
            <tr key={index}>
              <td style={{ whiteSpace: 'nowrap' }}>{student.name}</td>
              <td >{student.rollNumber}</td>
              <td>{student.cgpa}</td>
              <td>{student.backlogs}</td>
              <td>{student.branch}</td>
              <td style={{ whiteSpace: 'nowrap' }}>{student.languages.join(',
            ')}</td>
            </tr>
          ))}
        </tbody>
      </table>
    </div>
```

FULL STACK WEB DEVELOPMENT

```
);  
};  
export default StudentList;  
/* App.css */  
body {  
  font-family: 'Arial', sans-serif;  
  background-color: #f5f5f5;  
  margin: 0;  
  padding: 0;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  height: 100vh;  
}  
.container {  
  max-width: 400px;  
  width: 100%;  
  padding: 20px;  
}  
form {  
  background-color: #fff;  
  padding: 20px;  
  border-radius: 8px;  
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  
}  
label {  
  display: block;  
  margin-bottom: 8px;  
}  
input {  
  width: 100%;  
  padding: 4px;  
  margin-bottom: 16px;
```

FULL STACK WEB DEVELOPMENT

```
border: 1px solid #ccc;
border-radius: 4px;
}
h1, h2 {
  color: #333;
  text-align: center;
}
hr {
  border: 0;
  height: 1px;
  background-color: #ccc;
  margin: 20px 0;
}
.button-container {
  display: flex;
  justify-content: space-between;
  align-items: center; /* Center items vertically */
  margin-top: 20px; /* Add margin to separate from the form */
}
.button-link,
button {
  padding: 10px;
  text-align: center;
  text-decoration: none;
  color: #fff;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}
.button-link {
  background-color: #4caf50;
}
.button-link:hover {
```

FULL STACK WEB DEVELOPMENT

```
background-color: #45a049;
}
button {
  background-color: #008CBA;
}
button:hover {
  background-color: #0077a3;
}
table {
  border-collapse: collapse;
  width: 100%;
}
th, td {
  border: 1px solid #ddd;
  padding: 8px;
  text-align: left;
}
th {
  background-color: #4CAF50;
  color: white;
}
// App.js
import React, { useState } from 'react';
import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom';
import StudentForm from './StudentForm';
import StudentList from './StudentList';
import './App.css';
const App = () => {
  const [students, setStudents] = useState([]);
  const [minCGPA, setMinCGPA] = useState(0);
  const [eligibleStudents, setEligibleStudents] = useState([]);
  const addStudent = (student) => {
```

FULL STACK WEB DEVELOPMENT

```
    setStudents([...students, student]);
  };
const handleFilter = () => {
  const filtered = students.filter(
    (student) => student.cgpa >= minCGPA && student.backlogs === 0
  );
  setEligibleStudents(filtered);
};
const showAllDetails = eligibleStudents.length === 0;
return (
  <Router>
    <div className="container">
      <Routes>
        <Route
          path="/"
          element={
            <>
              <h1>Student Details</h1>
              <StudentForm onSubmit={addStudent} />
              <div className="button-container">
                <Link to="/view" className="button-link">View
Students</Link>
              </div>
            </>
          }
        />
        <Route
          path="/view"
          element={
            <>
              <StudentList students={students}
showAllDetails={showAllDetails} />
              <div>
```

FULL STACK WEB DEVELOPMENT

```
<br></br><label>Enter Minimum CGPA:</label>

<input
  type="number"
  step="0.01"
  value={minCGPA}
  onChange={(e) => setMinCGPA(parseFloat(e.target.value))}
/>

<button onClick={handleFilter}>Submit</button>
</div>

{eligibleStudents.length > 0 && (
  <div className="students">
    <h2>Eligible Students</h2>
    <table>
      <thead>
        <tr>
          <th>Name</th>
          <th style={{ whiteSpace: 'nowrap' }}>Roll
Number</th>
          <th>CGPA</th>
          <th style={{ whiteSpace: 'nowrap' }}>Languages
Known</th>
        </tr>
      </thead>
      <tbody>
        {eligibleStudents.map((student, index) => (
          <tr key={index}>
            <td style={{ whiteSpace: 'nowrap' }}>{student.name}</td>
            <td>{student.rollNumber}</td>
            <td>{student.cgpa}</td>
            <td style={{ whiteSpace: 'nowrap' }}>{student.languages.join(', ')}</td>
          </tr>
        ))}
      </tbody>
    </table>
  </div>
)}
```


FULL STACK WEB DEVELOPMENT

```
                </tbody>
            </table>
        </div>
    )}
</>
}
/>
</Routes>
</div>
</Router> );};
export default App;
```

Output:

Student Details

Name:

R,Anil Kumar

Roll Number:

068

CGPA:

7.5

Backlogs:

1

Branch:

Civil

Languages Known (comma-separated):

C,Python

Save and next

View Students

FULL STACK WEB DEVELOPMENT

All Students

Name	Roll Number	CGPA	Backlogs	Branch	Languages Known
K.Vishal Reddy	064	8.41	0	cse	c, py, java, dsa
R.Anil Kumar	068	7.5	1	Civil	C, Python

Enter Minimum CGPA:

Submit

Eligible Students

Name	Roll Number	CGPA	Languages Known
K.Vishal Reddy	064	8.41	c, py, java, dsa

(Even though Anil has more cgpa than minimum cgpa but he also has 1 backlog therefore he is not eligible.)