

1. Objective of the Day

The goal of Day 4 was to **start building the ReactJS frontend** for the Student Management System and **connect it with the backend API** developed using ASP.NET Core.

This includes setting up the React app structure, creating reusable components, integrating Bootstrap for styling, and testing the GET operation to fetch student data.

2. Topics Covered / Tasks Completed

1. Created React App:

- Initialized a new React project using the command:
- `npx create-react-app student-management-app`
- Installed dependencies:
- `npm install bootstrap react-router-dom`
- Configured the folder structure for better modularity:
- `student-management-app/`
 - `src/`
 - `components/`
 - `| └── StudentList.js`
 - `| └── AddStudentForm.js`
 - `| └── EditStudentForm.js`
 - `services/`
 - `| └── api.js`
 - `App.js`
 - `index.js`

2. Connected Bootstrap for Styling

- Added the Bootstrap import in `App.js`:
- `import "bootstrap/dist/css/bootstrap.min.css";`

3. Created API Service File

- Centralized the API base URL in `src/services/api.js`:

- const API_BASE_URL = "http://localhost:5205/api/Students";
- export default API_BASE_URL;

4. Developed Student List Component

- Created StudentList.js to fetch all students using the backend API.
- Implemented data fetching using the useEffect and fetch API.

3. Code Implemented

App.js

```
import React from "react";
import "bootstrap/dist/css/bootstrap.min.css";
import StudentList from "./components/StudentList";

function App() {
  return (
    <div className="container mt-4">
      <h2 className="text-center mb-4">Student Management System</h2>
      <StudentList />
    </div>
  );
}

export default App;
```

StudentList.js

```
import React, { useEffect, useState } from "react";
import API_BASE_URL from "../services/api";

function StudentList() {
  const [students, setStudents] = useState([]);
```

```
const fetchStudents = async () => {
  try {
    const response = await fetch(API_BASE_URL);
    const data = await response.json();
    setStudents(data);
  } catch (error) {
    console.error("Error fetching students:", error);
  }
};

useEffect(() => {
  fetchStudents();
}, []);

return (
<div>
  <h4 className="mb-3">Student List</h4>
  <table className="table table-bordered table-striped">
    <thead className="table-dark">
      <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Age</th>
        <th>Grade</th>
        <th>Course</th>
      </tr>
    </thead>
    <tbody>
```

```

{students.length > 0 ? (
  students.map((student) => (
    <tr key={student.id}>
      <td>{student.id}</td>
      <td>{student.name}</td>
      <td>{student.age}</td>
      <td>{student.grade}</td>
      <td>{student.courseName}</td>
    </tr>
  )))
  : (
    <tr>
      <td colSpan="5" className="text-center">
        No students found.
      </td>
    </tr>
  )}

</tbody>
</table>
</div>
);

}

export default StudentList;

```

api.js

```

const API_BASE_URL = "http://localhost:5205/api/Students";
export default API_BASE_URL;

```

4. Output Verification

After starting both the backend (dotnet run) and frontend (npm start), the student data successfully appeared on the browser from the SQL Server database.

Working Flow:

1. The backend API runs on port **5205** (ASP.NET Core).
2. The React app runs on port **3000**.
3. Data fetched using the fetch API call from the backend.
4. Displayed in a clean, responsive table layout styled with Bootstrap.

5. Challenges Faced

- Faced **CORS policy** issue initially. Resolved it by adding CORS configuration in backend Program.cs:

```
builder.Services.AddCors(options =>
{
    options.AddPolicy("AllowAll", builder =>
        builder.AllowAnyOrigin().AllowAnyHeader().AllowAnyMethod());
});
```

- Data initially not loading due to API URL typo (extra slash). Corrected base URL.