Project:

useForm():

- ✓ useForm() is custom hook for managing forms easily. It takes optional arguments.
- ✓ we can easily validate the form by using useForm()
- ✓ we need to install react-hook-form by using below command

npm i react-hook-form

Student.js

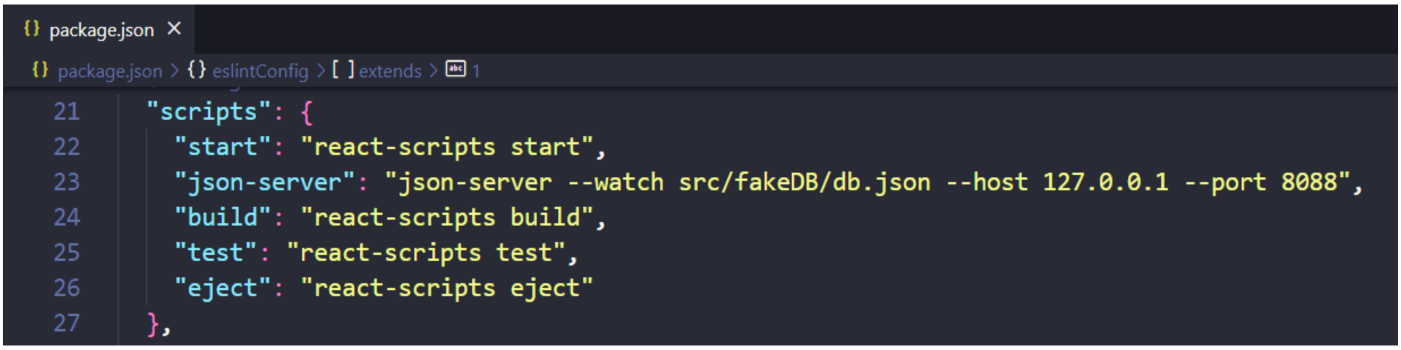
```
import { useForm } from "react-hook-form";
import { saveUser } from "../FakeDB/DBService";

const Student=()=>{

    const {register,handleSubmit}=useForm()

    function getRegisterStudent(studObj){

        console.log(studObj);
        axios.post("http://127.0.0.1:8090/student",studObj)
        reset()
  }

    return (<>
     <form onSubmit={handleSubmit(getRegisterStudent)}>

        <div className="row">
        <div className="col-md-3">
            <h1 className="text-primary text-center">Register Student</h1>
            <div class="mb-3">
                <input type="text" class="form-control" placeholder="Firstname"
{...register("firstname")}/>
            </div>
            <div class="mb-3">
                <input type="text" class="form-control"  placeholder="Lastname"
{...register("lastname")}/>
            </div>
            <div class="mb-3">
                <input type="text" class="form-control" placeholder="Mobile"
{...register("mobile")}/>
            </div>
            <div class="mb-3">
                <input type="text" class="form-control"  placeholder="Email"
{...register("email")}/>
            </div>
            <div class="mb-3 text-center">
                <input type="submit" class="btn btn-success"/>
            </div>
            </div>
        </div>
    </form>
    </>)
}
export default User;
```

now, we need to store this Object, so we will create fake database by using JSON File.
step 1: create a **FakeDB** folder
step 2: create a **db.json** file in **FakeDB** folder
e.g: db.json

```
{
  "student": [
    {
      "id": 1,
      "firstname": "Rakesh",
      "lastname": "Kolhe",
      "mobile": 9865743210,
      "email": "rk@gmail.com"
    },
    {
      "id": 2,
      "firstname": "Yogesh",
      "lastname": "Patil",
      "mobile": 9865743211,
      "email": "yp@gmail.com"
    }
      ]
}
```

- ✓ We want to run this database on server, so we need to install **json-server**.
- ✓ To install json-server : **npm i json-server**
- ✓ After installation Configure json server path with database file in **package.json** file :

"json-server": "json-server --watch src/FakeDB/db.json --host 127.0.0.1 --port 8088",



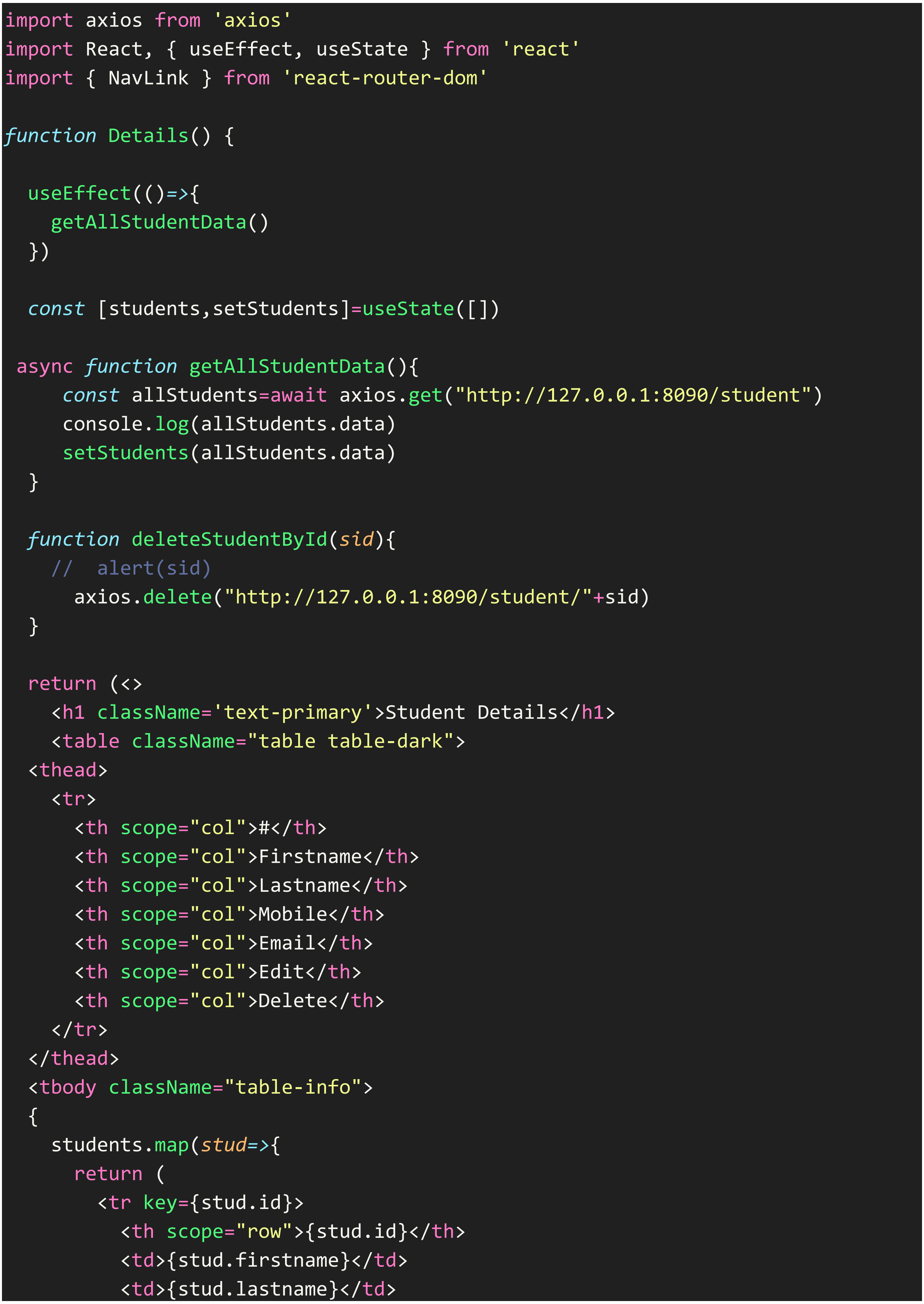
```
  "scripts": {
    "start": "react-scripts start",
    "json-server": "json-server --watch src/fakeDB/db.json --host 127.0.0.1 --port 8088",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
```

To Run JSON-Server: **npm run json-server**

To communicate with our fake database, we need to use axios Library.

- ✓ Axios is a JavaScript library that allows you to connect with the backend API and manage requests made via the HTTP protocol.
- ✓ To install axios library :**npm install axios**

Details.js



```
import axios from 'axios'
import React, { useEffect, useState } from 'react'
import { NavLink } from 'react-router-dom'

function Details() {

  useEffect(()=>{
    getAllStudentData()
  })

  const [students,setStudents]=useState([])

 async function getAllStudentData(){
     const allStudents=await axios.get("http://127.0.0.1:8090/student")
     console.log(allStudents.data)
     setStudents(allStudents.data)
  }

  function deleteStudentById(sid){
    //  alert(sid)
      axios.delete("http://127.0.0.1:8090/student/"+sid)
  }

  return (<>
    <h1 className='text-primary'>Student Details</h1>
    <table className="table table-dark">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">Firstname</th>
      <th scope="col">Lastname</th>
      <th scope="col">Mobile</th>
      <th scope="col">Email</th>
      <th scope="col">Edit</th>
      <th scope="col">Delete</th>
    </tr>
  </thead>
  <tbody className="table-info">
  {
    students.map(stud=>{
      return (
        <tr key={stud.id}>
          <th scope="row">{stud.id}</th>
          <td>{stud.firstname}</td>
          <td>{stud.lastname}</td>
          <td>{stud.mobile}</td>
          <td>{stud.email}</td>
          <td>
              <NavLink to={`/editStudent/${stud.id}`}>     {/* Step 1 */}
                <i class="bi bi-pencil-square text-primary"></i>
              </NavLink>
          </td>
          <td>
              <i class="bi bi-trash text-danger"
onClick={()=>{deleteStudentById(stud.id)}}></i>
          </td>
        </tr>
      )
    })
  }
  </tbody>
</table>
</>)
}

export default Details
```

EditStudent.js

```
import axios from 'axios'
import React, { useEffect, useState } from 'react'
import { useForm } from 'react-hook-form'
import { useNavigate, useParams } from 'react-router-dom'

function EditStudent() {            //step 2 creating Edit Component
   const {sid}=useParams()      // Step 4

    useEffect(()=>{
        getStudentById()        //step 6
    })

    const [stud,setStudent]=useState({})    // step 8

    const {register,setValue,handleSubmit}=useForm()     // step 10
    const navi=useNavigate()

    async function getStudentById(){           //step 7 --> async await
      //  http://127.0.0.1:8090/student/2
   const student= await axios.get("http://127.0.0.1:8090/student/"+sid)//Step 5
      console.log(student.data)
      setStudent(student.data)         // step 8
    }
    function getUpdatedData(studOBJ){
        //http://127.0.0.1:8090/student /2 , studOBJ
        axios.put("http://127.0.0.1:8090/student/"+studOBJ.id,studOBJ)
        console.log(studOBJ)
        navi("/details")
    }

  return (<>
        <div className='row'>               {/* Step 9 Bootstrap Form */}
        <h3 className='text-primary mt-3'>Update Student
Profile</h3>
        <div className='col-md-3'>
          <form onSubmit={handleSubmit(getUpdatedData)}>
            <div>
              <input type="hidden" {...setValue("id",stud.id)}
{...register("id")} />
            </div>
          <div class="mb-3">          {/* Step 11 */}
            <input type="text" class="form-control" placeholder="Firstname"
{...setValue("firstname",stud.firstname)} {...register("firstname")}/>
          </div>
          <div class="mb-3">
            <input type="text" class="form-control" placeholder="Lastname"
{...setValue("lastname",stud.lastname)} {...register("lastname")}/>
          </div>
          <div class="mb-3">
            <input type="text" class="form-control" placeholder="Mobile"
{...setValue("mobile",stud.mobile)} {...register("mobile")}/>
          </div>
          <div class="mb-3">
            <input type="text" class="form-control" placeholder="Email"
{...setValue("email",stud.email)} {...register("email")}/>
          </div>
         <div class="mb-3 text-center">
          <input type="submit" class="btn btn-outline-success" value="Update"/>
         </div>
         </form>
        </div>
    </div>
   </>)
}

export default EditStudent
```

FormValidation:

formState : This object contains information about the entire form state. It helps you to keep on track with the user's interaction with your form application.

errors: is an object with errors field.

To read error message of field : {errors.fieldname?.message}

Student.js

```
import axios from 'axios';
import React from 'react'
import { useForm } from 'react-hook-form'

function Student() {

 const {register,handleSubmit,reset,formState:{errors}}=useForm()

 function getRegisterStudent(studObj){
    console.log(studObj);
    axios.post("http://127.0.0.1:8090/student",studObj)
    reset()
 }
 return (
  <div className='row'>
    <div className='col-md-3'>
     <h1 className='text-primary'>Register Student</h1>
     <form onSubmit={handleSubmit(getRegisterStudent)}>
     <div class="mb-3">
      <input type="text" class="form-control" placeholder="Firstname"
{...register("firstname",
      {required:"This field is Required!"}) } />
      <h6>{errors.firstname?.message}</h6>
     </div>
     <div class="mb-3">
      <input type="text" class="form-control" placeholder="Lastname"
{...register("lastname",
      {required:"This field is Required!" ,
       minLength:{
         value:3,
         message:" Min 3 chars Allowed!"
       },
       maxLength:{
        value:15,
        message:"Max 15 chars Allowed!"
       }
      })} />
      <h6>{errors.lastname?.message}</h6>
     </div>
     <div class="mb-3">
```

```
            <input type="text" class="form-control"
placeholder="Mobile" {...register("mobile",
            {required:"This field is Required!",
              pattern:{
                  value:/^[6789]{1}[0-9]{9}$/,
                  message:"Invalid Mobile Number!"
              }
            } )} />
            <h6>{errors.mobile?.message}</h6>
          </div>
          <div class="mb-3">
            <input type="text" class="form-control" placeholder="Email"
{...register("email")} />
          </div>
          <div class="mb-3 text-center">
            <input type="submit" class="btn btn-outline-success" />
          </div>
          </form>
        </div>
    </div>
  )
}
export default Student
```