

Step1: Project and Folder

```
C:\demo\jsmarch24\L21>mkdir stu_project
C:\demo\jsmarch24\L21>cd stu_project
C:\demo\jsmarch24\L21\stu_project>mkdir server
```

Step2: React JS

S1:

```
C:\demo\jsmarch24\L21\stu_project>mkdir client
C:\demo\jsmarch24\L21\stu_project>cd client
C:\demo\jsmarch24\L21\stu_project\client>npx create-react-app stuapp
C:\demo\jsmarch24\L21\stu_project\client>cd stuapp
C:\demo\jsmarch24\L21\stu_project\client\stuapp>npm install formik axios
C:\demo\jsmarch24\L21\stu_project\client\stuapp>cd src
C:\demo\jsmarch24\L21\stu_project\client\stuapp\src>notepad Student.js
C:\demo\jsmarch24\L21\stu_project\client\stuapp\src>cd..
C:\demo\jsmarch24\L21\stu_project\client\stuapp>npm start
```

S2:

Student.js

```
import { useFormik } from "formik";
import axios from "axios";

export default function Student()
{
    const initialValues = {"rno":"","name":"","marks":""};

    const validate = (values) => {
        const errors = {};
        if (values.rno === "")
            errors.rno = "rno cannot be empty";
        if (parseInt(values.rno) < 1)
            errors.rno = "rno shud be min 1";

        if (values.name === "")
            errors.name = "name cannot be empty";
        if (! values.name.match(/^[A-Za-z ]+$/))
            errors.name = "name shud contain only alphabets";

        if (values.marks === "")
            errors.marks = "marks cannot be empty";
        if ( ( parseInt(values.marks) < 0 ) || ( parseInt(values.marks) > 100 ) )
            errors.marks = "marks shud be between 0 to 100";

        return errors;
    }
}
```



```

const onSubmit = (values, {resetForm}) => {
  let r = values.rno;
  let n = values.name;
  let m = values.marks;
  let data = {r, n, m};
  let url = "http://localhost:9000/save";
  axios.post(url, data)
    .then(res => {
      alert("record saved");
      formik.resetForm({values: ""});
    })
    .catch(err => alert("issue " + err));
}

const formik = useFormik({initialValues, validate, onSubmit});

return(
  <>
  <center>
    <h1> Student App </h1>
    <form onSubmit={formik.handleSubmit}>
      <input type="number" placeholder="enter rno"
        name="rno" onChange={formik.handleChange} value={formik.values.rno}/>
      { formik.errors.rno ? <div className="err"> { formik.errors.rno } </div> : ""}
      <br/><br/>
      <input type="text" placeholder="enter name"
        name="name" onChange={formik.handleChange} value={formik.values.name}/>
      { formik.errors.name ? <div className="err"> { formik.errors.name } </div> : ""}
      <br/><br/>
      <input type="number" placeholder="enter marks"
        name="marks" onChange={formik.handleChange} value={formik.values.marks}/>
      { formik.errors.marks ? <div className="err"> { formik.errors.marks } </div> : ""}

      <br/><br/>
      <input type="submit" value="Save" />
      <br/><br/>
    </form>
  </center>
  </>
);
}

```


App.js

```
import logo from './logo.svg';
import './App.css';
import Student from './Student';
```

```
function App() {
  return (
    <Student/>
  );
}
export default App;
```

App.css

```
*
{
  font-size:3vw;
}
body
{
  background-color:lightblue;
}
.err
{
  color:red;
}
```


Step3: Express

S1:

```
C:\demo\jsmarch24\L21\stu_project\server>npm install express cors nodemailer
C:\demo\jsmarch24\L21\stu_project\server>notepad index.js
C:\demo\jsmarch24\L21\stu_project\server>nodemon index.js
```

S2:

index.js

```
const express = require("express");
const cors = require("cors");
const nodemailer = require("nodemailer");

const app = express();
app.use(cors());
app.use(express.json());

app.post("/save", (req, res) => {
  let data = [req.body.r, req.body.n, req.body.m]
  let name = req.body.n;
  let txt = "rno = " + req.body.r + " name = " + req.body.n + " marks = " + req.body.m;

  let transporter = nodemailer.createTransport({
    service: "gmail",
    auth: {
      user: "tester18april24@gmail.com",
      pass: "sfrfslgqgsokkcqb"
    }
  })

  let mailOptions = {
    from: "tester18april24@gmail.com",
    to: "classeskamalsir@gmail.com",
    subject: "Info of " + name,
    text: txt
  }

  transporter.sendMail(mailOptions, (err, info) => {
    if (err) {
      console.log("mail err ", err);
      res.status(500).json(err);
    }
    else {
      console.log("mail send", info.response);
      res.status(200).json("mail send");
    }
  })
});

app.listen(9000, () => { console.log("ready to serve @ 9000");})
```