**Node-react app**

For this to make it more orderly create folder „web-app“ and inside that folder create two new folders one you can name server(node) and second u can name client(react). In server folder we are gonna have backend side of our application where we will first install express.js:

-npm i express

With this command we will get node-modules for our server side.Then we install nodemon for our server to update automatically if we make any changes to our server:

-npm i nodemon -D

After that we will need package.json file:

-npm init -y

Inside package.json file in „main:“ we will write name of our js file in which we are gonna do all coding. Default name is index.js but we changed it to server.js.

  "main": "server.js",

After that we create server.js file. Then we return to json file and add this to „scripts:“ object:

    "start": "node server.js",

    "dev": "nodemon server.js --ignore client"

After all that setup lets get into server.js. Inside we have to call express library:

const express = require('express')

const app = express();

Now that we have access to express library lets create server for our server side:

app.listen(5000, () => { console.log("Server started at port 5000!") })

With this we created server on localhost:5000.

Now we have to create GET method that will send files to frontend(react) side of our app.

//With get request we send data to frontend

app.get("/api", (req,res) => {

    let rawdata = fs.readFileSync('cats.json');

    cats = JSON.parse(rawdata);

    res.json(cats)

})

In here we read json object from json file „cats.json“ and with response „res“ we send data to frontend. To see json write localhost:5000/api in browser.

Now that we are done with that we need to switch to client(react) side of our app.

Inside client folder we need to install react app:

     npx create-react-app .

After the installation go in src folder where u can delete everything except index.js and app.js. Than inside package.json file we have to add at the and proxy to our server side:

  },

  "proxy": "http://localhost:5000"

}

Thats it for json file now we get into app.js. Inside delete all unnecessary code u dont need, basicly everything inside div element. After that we want to fetch data we sent from our server side. For this we will need useState and use Effect functions from react library. First we declare useState function where we are gonna save data. After that we declare useEffect function where we are gonna use fetch function.

    let [cats, getNodeData] = useState([{}])

    //Geting data that is send from backend/node/express in this case object cats

    useEffect(() => {

      fetch("/api").then(

        resp => resp.json()

      ).then(

        data => {

          getNodeData(data)

        }

      )

    }, [])

With fetch we are fetching response and data from '/api' and saving that data into 'cats' json object. Now all data that we send with 'api' route is saved into cats object and we can access it however we want.

Now if we wanna send data from react to node for lets say add new object to our file or delete existing object in file, for this we are gonna need POST method. First of all for this we are gonna need certain librarys for react side we will have to install 'axios':

npm install axios

import axios from 'axios'

After installing it we have to import it, his function is for POST method.

    //Sending new cat that we created to backend side

    const toggleInsert = () => {

      axios.post("http://localhost:5000/Addcat", newCatObj)

        .then(resp => {

          console.log(resp)

        })

        .catch(error => {

          console.log(error)

    })};

Inside axios.post function we declare where we wanna send our data in this case: 'http://localhost:5000/Addcat' and object we wanna send: newCatObj. Now we have to switch to server side to our server.js. First we wanna install cors library.

npm install cors

const cors = require('cors');

//use cors to allow cross origin resource sharing otherwise its blocked

app.use(

    cors({

      origin: 'http://localhost:3000',

      credentials: true,

    }),

);

Without corse we cant send data to server side from client side but cors fixes that.

Also we need these two lines of code otherwise our POST return is undefined:

app.use(express.json());

app.use(express.urlencoded({ extended: false }));

After all that we can start creating our post function, same thing like GET in function we define route: '/Addcat' and function for request and response but for POST we will only gonna need request. To access sent object we can create new object where we are gonna save all out data. Lets say we wanna access name we wanna write: req.body.key which means from request inside body get name we sent.

//With post we get data from frontend

app.post("/Addcat", function (req, res) {

    let newCat = {

        key: req.body.key,

        name: req.body.name,

        age: req.body.age,

        color: req.body.color

    }

    cats.push(newCat)

Now that we created that new object we can add it to json object with cats.push(->objectName<-) after that we save it to json file.

    let jsonContent = JSON.stringify(cats, null, 2);

    fs.writeFile("cats.json", jsonContent, 'utf8', () => {});