

Routes->src->user

UserController.js

```
var userService = require('./userService');

var getDataConntrollerfn = async (req, res) =>
{
    var empolyee = await userService.getDataFromDBService();
    res.send({ "status": true, "data": empolyee });
}

var createUserControllerFn = async (req, res) =>
{
    var status = await userService.createUserDBService(req.body);
    if (status) {
        res.send({ "status": true, "message": "User created successfully" });
    } else {
        res.send({ "status": false, "message": "Error creating user" });
    }
}

var updateUserController = async (req, res) =>
{
    console.log(req.params.id);
    console.log(req.body);
    var result = await userService.updateUserDBService(req.params.id, req.body);
    if (result) {
        res.send({ "status": true, "message": "User Updateeeeeeddddd" });
    } else {
        res.send({ "status": false, "message": "User Updateeeeeeddddd Faileddddddd" });
    }
}
```

```

var deleteUserController = async (req, res) =>
{
  console.log(req.params.id);
  var result = await userService.removeUserDBService(req.params.id);
  if (result) {
    res.send({ "status": true, "message": "User Deletedddd" });
  } else {
    res.send({ "status": false, "message": "User Deletedddd Faileddddddd" });
  }
}

module.exports = { getDataControllerfn,
createUserControllerFn,updateUserController,deleteUserController };

```

userModel.js

```

var mongoose = require('mongoose');
var Schema = mongoose.Schema;
var userSchema = new Schema({
  name: {
    type: String,
    required: true
  },
  address: {
    type: String,
    required: true},
  phone: {
    type: String,
    required: true}}});
module.exports = mongoose.model('employees', userSchema);

```

userService.js

```
return new Promise(function checkURL(resolve, reject) {
  userModel.find({}, function returnData(error, result) {
    if (error) {
      reject(false);
    } else {
      resolve(result);
    }
  });
});

module.exports.createUserDBService = (userDetails) => {
  return new Promise(function myFn(resolve, reject) {
    var userModelData = new userModel();
    userModelData.name = userDetails.name;
    userModelData.address = userDetails.address;
    userModelData.phone = userDetails.phone;

    userModelData.save(function resultHandle(error, result) {
      if (error) {
        reject(false);
      } else {
        resolve(true);
      }
    });
  });
};

module.exports.updateUserDBService = (id,userDetails) => {
  console.log(userDetails);
  return new Promise(function myFn(resolve, reject) {
```

```

    userModel.findByIdAndUpdate(id,userDetails, function returnData(error, result) {
      if(error)
      {
        reject(false);
      }
      else
      {
        resolve(result);
      }
    });
  });
}

module.exports.removeUserDBService = (id) => {
  return new Promise(function myFn(resolve, reject) {
    userModel.findByIdAndDelete(id, function returnData(error, result) {
      if(error)
      {
        reject(false);
      }
      else
      {
        resolve(result);
      }
    });
  });
}

```

routes->routes.js

```
var userModel = require('./userModel');

module.exports.getDataFromDBService = () => {

var express = require('express');

const router = express.Router();

var userController = require('./src/user/userController');

router.route('/user/getAll').get(userController.getDataControllerfn);

router.route('/user/create').post(userController.createUserControllerFn);

router.route('/user/update/:id').patch(userController.updateUserController);

router.route('/user/delete/:id').delete(userController.deleteUserController);

module.exports = router;
```

package.json

```
{

  "name": "wadproject",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

    "test": "echo \\\"Error: no test specified\\\" && exit 1"

  },

  "author": "",

  "license": "ISC",

  "dependencies": {

    "cors": "^2.8.5",

    "express": "^4.18.2",

    "mongoose": "^6.7.5",

    "routes": "^2.1.0"  }}
```

Server.js

```
var express = require('express');
var server = express();
var routes = require('./routes/routes');
var mongoose = require('mongoose');
const cors = require('cors');

mongoose.connect("mongodb://127.0.0.1:27017/EST",{useNewUrlParser: true,
useUnifiedTopology: true },function checkDB(error)
{
    if(error)
    {
        console.log("errorr")
    }
    else
    {
        console.log("DB Connecteddddd!!!!!!!!!!!!")
    }
});

server.use(express.json());
server.use(cors());
server.use(routes);
server.listen(8000,function check(error)
{
    if(error)
    {
        console.log("errorr")
    }
    else
```

```
{
  console.log("starteddddddd")
}
});
```

frontend-app

app.js

```
import 'bootstrap/dist/css/bootstrap.min.css';
import EmployeeCrud from './components/EmployeeCrud';
function App() {
  return (
    <EmployeeCrud/>
  );
}
export default App;
```

src->components->EmployeeCrud.js

```
import axios from 'axios';
import {useEffect,useState } from "react";

function EmployeeCrud() {

  const [_id, setId] = useState("");
  const [name, setName] = useState("");
  const [address, setAddress] = useState("");
  const [phone, setMobile] = useState("");
  const [employees, setUsers] = useState([]);
```

```
useEffect(() => {  
  (async () => await Load())();  
}, []);
```

```
async function Load() {  
  const result = await axios.get("http://localhost:8000/user/getAll");  
  setUsers(result.data.data);  
  console.log(result.data);  
}
```

```
async function save(event) {  
  event.preventDefault();  
  try {  
    await axios.post("http://localhost:8000/user/create", {  
      name: name,  
      address: address,  
      phone: phone,  
    });  
    alert("Employee Registration Successfully");  
    setId("");  
    setName("");  
    setAddress("");  
    setMobile("");  
    Load();  
  } catch (err) {  
    alert("User Registration Failed");  
  }  
}
```



```

}

async function editEmployee(employees) {
  setName(employees.name);
  setAddress(employees.address);
  setMobile(employees.phone);

  setId(employees._id);
}

```

```

async function DeleteEmployee(_id) {
  await axios.delete("http://localhost:8000/user/delete/" + _id);
  alert("Employee deleted Successfully");
  Load();
}

```

```

async function update(event) {
  event.preventDefault();
  try {
    await axios.patch(
      "http://localhost:8000/user/update/" +
      employees.find((u) => u._id === _id)._id || _id,
      {
        _id: _id,
        name: name,
        address: address,
        phone: phone,
      }
    );
  }
}

```

```
    alert("Registration Updateddddd");  
    setId("");  
    setName("");  
    setAddress("");  
    setMobile("");  
    Load();  
  } catch (err) {  
    alert(err);  
  }  
}
```