Job Finder

1. Npm init

Serve pra iniciar o projeto

1. Criar app.js

O arquivo principal do projeto

1. Npm install(express express-handlebars body-parser sequelize [sqlite3@5.0.0](mailto:sqlite3@5.0.0))

Instalar as bibliotecas necessárias

1. Npm install -D nodemon

Instalar o nodemon

1. No app .js:

const express = require("express")

const app = express()

const PORT = 3000

app.listen(PORT, function() {

    console.log(`O express está rodando na porta ${PORT}`)

})

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

    res.send("Está funcionando")

})

Verifica se o projeto está funcionando

1. No package.json:

"scripts": {

    "dev": "nodemon app.js"

  },

Adciona o nodemon q permite atualizar os codigos pro localhost sozinho usando npm run dev

1. Criar pasta db

Pasta que terá tudo sobre o banco de dados

1. Dentro do db criar o arquivo connection.js

Que conectará com o banco de dados

1. No connection:

const Sequelize = require("sequelize")

const sequelize = new Sequelize({

    dialect: "sqlite",

    storage: "./db/app.db"

})

module.exports = sequelize

Vai conectar o banco de dados sqlite e exportar

1. No app.js:

const db = require("./db/connection.js")

vai se conectar com o connection

1. No app.js antes do app.get

Db

.authenticate()

.then(() => {

    console.log("Conectou ao banco com sucesso")

})

.catch (err => {

    console.log("Ocorreu um erro ao conectar")

})

Testar a conexão com o banco de dados

1. Instalar pelo navegador o “db browserfor SQLite standart installer for 64-bit Windows”

não precisa assinalar nada

1. Abrir o arquivo deo app.db
2. Criar tabela com id(integer, Primary, AutiIncrement)

Criar text(text, salary, company, email, createdAt, updatedAt)  
Criar integer (new)´

1. Cria uma pasta chamado models, criar um arquivo job.js e add

const Sequelize = require("sequelize")

const db = require("../db/connection")

const Job = db.define("job", {

    title: {

       type: Sequelize.STRING,

      },

      description: {

        type: Sequelize.STRING

      },

      salary: {

        type: Sequelize.STRING

      },

      company: {

        type: Sequelize.STRING

      },

       email: {

        type: Sequelize.STRING

      },

      new\_job: {

        type: Sequelize.INTEGER

      }

})

module.exports = Job

1. Criar pasta routes e add um arquivo jobs.js
2. No jobs.js

const express = require("express")

const router  = express.Router

const Job     = require("../models/Job")

router.post("/add", (req, res) => {

    let {title,  salary, company, description, email, new\_job} = req.body

})

1. No app.js

const bodyParse = require('body-parser')

1. Embaixo do app.listen:

app.use(bodyParse.urlencoded({extended: false}))

1. No jobs.js embaixo do let

Job.create({

        title,

        salary,

        company,

        description,

        email,

        new\_job

    }).then(() => res.redirect("/"))

    .catch(err => console.log(err))

module.exports = router

Inserindo os dados

1. No app.js

app.use("/jobs", require("./routes/jobs"))

1. No postman

Muda de get pra post e coloca o link “localhost:3000/jobs/add”

1. No postman vai em header e colocar key “Content-Type” com o value de “application/json”
2. Vai em body e no x-www-form-unlencoded coloca a

key “title” value “Dev PHP”

key “description” value “O programador deve trabalhar com Lavarel e Vue”

key “company” value “Devs Hub”

key “salary” value “5000”

key “email” value “matheus@devshub.com”

key “new\_job” value “1”

E clica em send

1. Criar index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Job Finder</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

    <link rel="stylesheet" href="public/css/styles.css">

</head>

<body>

    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"

        integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r"

        crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"

        integrity="sha384-0pUGZvbkm6XF6gxjEnlmuGrJXVbNuzT9qBBavbLwCsOGabYfZo0T0to5eqruptLy"

        crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"

        crossorigin="anonymous"></script>

</body>

</html>

1. Criar pasta public, dentro dela a pasta css, e dentro da pasta css cria styles.css