**TODO LIST**

**Create models folder**

**Under models folder create**

**INDEX.JS**

const mongoose = require("mongoose")

mongoose.connect("mongodb+srv://hasini:Autogenerated@cluster0.a7oxyh6.mongodb.net/?retryWrites=true&w=majority", {

  // connecting to the mongodb database name: "todo-app" locally

  keepAlive: true, // keeping the connection alive

  useNewUrlParser: true,

  useUnifiedTopology: true,

}).then(()=>{

    console.log("Database Connection")

}

).catch(err=>{

    console.log("Database error"+ err)

})

mongoose.set("debug", true) // enabling debugging information to be printed to the console for debugging purposes

mongoose.Promise = Promise // setting mongoose's Promise to use Node's Promise

module.exports.Todo = require("./todolist") // requiring the todo model that we just created in mongodb

**TODOLIST.JS**

const mongoose = require("mongoose") // requiring the mongoose package

const todoSchema = new mongoose.Schema({

  //id:mongoose.Types.ObjectId,

  // creating a schema for todo

  Description: {

    // field1: task

    type: String, // task is a string

  },

  Completed: {

    // field2: completed

    type: Boolean, // it is a boolean

  },

  Deleted: {

    // field2: completed

    type: Boolean, // it is a boolean

  },

})

const todoModel = mongoose.model("TodoList", todoSchema) // creating the model from the schema

module.exports = todoModel // exporting the model

**ROUTES**

**Under Routes.js**

const express = require("express") // our express server

const router = express() // generate an app object

const Todo = require("../models/todolist")

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

    try

    {

        const doLists = await Todo.find()

        res.json(doLists)

    }

    catch(err)

    {

        res.status(500).json({message : err.message})

    }

})

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

    try {

      const doLists = await Todo.create(req.body)

      return success(res, doLists)

    }

    catch (err) {

      res.status(500).json ({message: "failed to create todos" })

    }

  })

  router.put("/todos/:id", async (req, res, next) => {

    try {

      const todo = await Todo.findByIdAndUpdate(req.params.id, req.body, {

        new: true,

      })

      return success(res, todo)

    } catch (err) {

      next({ status: 400, message: err+ "failed to update todo" })

    }

  })

  router.delete("/todos/:id", async (req, res, next) => {

    try {

      await Todo.findByIdAndRemove(req.params.id)

      return success(res, "todo deleted!")

    } catch (err) {

      next({ status: 400, message: err +"failed to delete todo" })

    }

  })

  function success(res, payload) {

    return res.status(200).json(payload)

  }

  router.use((err, req, res, next) => {

    return res.status(err.status || 400).json({

      status: err.status || 400,

      message: err.message || "there was an error processing request",

    })

  })

module.exports=router

**SERVER.JS**

const express = require("express") // our express server

const app = express() // generate an app object

const bodyParser = require("body-parser") // requiring the body-parser

const PORT = process.env.PORT || 3000 // port that the server is running on => localhost:3000

app.use(bodyParser.json()) // telling the app that we are going to use json to handle incoming payload

var cors = require('cors')

const db =require("./models")

var corsOptions = {

  origin: '\*',

  optionsSuccessStatus: 200 // some legacy browsers (IE11, various SmartTVs) choke on 204

}

app.use(cors())

const routesub = require("./routes/routes")

app.use('/todolist',routesub)

app.listen(PORT, () => {

  // listening on port 3000

  console.log(`listening on port ${PORT}`) // print this when the server starts

})