Paste bootstrap css link in head – in index.html

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

Paste script bootstrap js link in bottom of body- in index.html

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

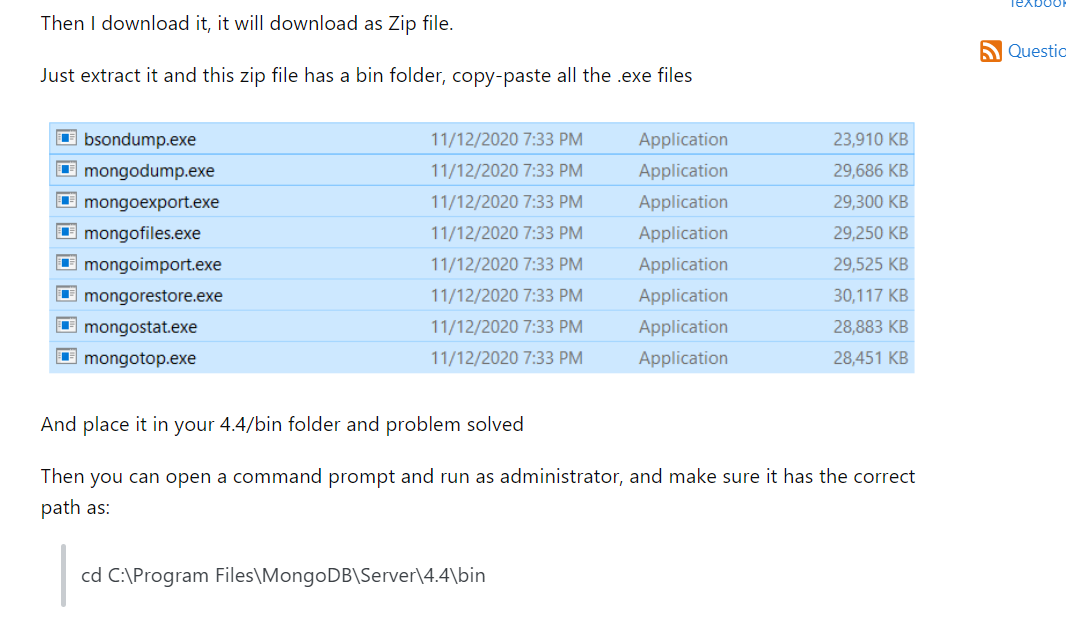
Now u can use all bootstrap navbar,search etc . in ur components App.js, Navbar.js etc..

In video 4: while importing mongodb data – using mongodbimport command in cmd –

At that point,

<https://www.mongodb.com/try/download/database-tools>

install this mongodb command line tools.



Extract all these files from that downloaded file and paste it in path c:/programfiles/mongodb/server/7.0/bin

Now execute mongodbimport command in cmd – it will work out.

Using mongodbimport import json data fooditems,foodcategory – into gofood collection

**Db.js**

Const mongoose =require(‘mongoose’);

Const mongodb=await Mongoose.connect(‘’).then(()=> console.log(‘db connected’))

//Mongoose.connect doesn’t accept callback so used .then

Module.exports=mongodb;

**Index.js**

Const express=require(‘express’);

Const mongodb=require(‘./db’);

mongoDB();

In video 5 – fetchdata= find operation – use await

Creating backend data.

Create backend named folder , do npm init

Create index.js – express

**To access an existing collection – use mongoose.connection.db.collection**

const mongoose = require('mongoose');

const MongoURI = 'mongodb+srv://zomatozee:zomatozee@cluster0.xlrs6ke.mongodb.net/zomatodb?retryWrites=true&w=majority'

const mongoDB = async () => {

    await mongoose.connect(MongoURI).then(async () => {

        console.log('db connected')

        //fetching data

        const fetchedData = await mongoose.connection.db.collection('fooditems');

        const data= await  fetchedData.find({}).toArray();

        const foodCategory=await mongoose.connection.db.collection('foodCategory');

        const catData=await foodCategory.find({}).toArray();

            if(data){

                global.fooditems=data;

                global.foodCategory=catData;

                // console.log(global.fooditems);

                console.log(global.foodCategory)

            }

            else{

                console.log(e)

            }

It returning a promise.

    // console.log(`fetched data is ${fetchedData}`) // u get output as object

    // console.log('fetched data is:', fetchedData.find({}).toArray() ) // u get output as it returning a promise - so add await.

    console.log('fetched data is:', await fetchedData.find({}).toArray() )

**db.js :**

const mongoose=require('mongoose');

const MongoURI='mongodb+srv://zomatozee:zomatozee@cluster0.xlrs6ke.mongodb.net/zomatodb?retryWrites=true&w=majority'

const mongoDB=async()=>{

    await mongoose.connect(MongoURI).then(async()=>{

    console.log('db connected')

    const fetchedData=await mongoose.connection.db.collection('fooditems');

    // console.log(`fetched data is ${fetchedData}`)

    // console.log('fetched data is:', fetchedData.find({}).toArray() ) // u get output as it returning a promise - so add await.

    console.log('fetched data is:', await fetchedData.find({}).toArray() )

}).catch((e)=>{

    console.log('error in connnecting db',e)

})

}

module.exports= mongoDB;

here if I try to Insert different coloumn key – value pairs in mongodb compass fooditems,foodcollections, it get accepted.

So to define strict schema and model

**Fetching data from mongodb :**

In video 7 – use express validator version 6.15.0

<https://express-validator.github.io/docs/6.15.0/>

in video 9 –

const mongoose=require('mongoose');

const MongoURI='mongodb+srv://zomatozee:zomatozee@cluster0.xlrs6ke.mongodb.net/zomatodb?retryWrites=true&w=majority'

const mongoDB=async()=>{

    await mongoose.connect(MongoURI).then(async()=>{

    console.log('db connected')

    const fetchedData=await mongoose.connection.db.collection('fooditems');

    // console.log(`fetched data is ${fetchedData}`)

    // console.log('fetched data is:',  fetchedData.find({}).toArray() ) // u get output as it returning a promise - so add await.

    //  await fetchedData.find({}).toArray((e,data)=>{

    //     if(e) console.log(e);

    //     else{

    //         global.fooditems=data;

    //         console.log(`data is :${global.fooditems}`)

    //     }

    // })

    const data=await fetchedData.find({}).toArray();

    if(data) {

        global.fooditems=data;

        // console.log(data)

        console.log(global.fooditems)

    }

    else{

        console.log('error in fetching db')

}

    }

    )}

module.exports= mongoDB;