**Mongoose instructions @ 11 June 2019**

npm install mongoose

Add to index.js

const mongoose = require('mongoose')

Add to index.js, open a conncetion to database in index.js. connect function will raise an error exception/event if for some reasons it couldn’t connect

mongoose.connect('mongodb://localhost/contacts', {

useNewUrlParser: true

})

extract a connection after connected. We are capturing an event here and pass a callback to accept the error

mongoose.connection.on("error", error => console.log(error))

*Establish a connection and establish error handling if connection fails*

Mongoose works by creating models. TO create models, have to define the structure of those models and we do that with schema in mongoose.

Create new folder > models

Add contact\_model.js to models folder

Create a schema in contact\_model.js with the fields from the contact form

const ContactSchema = new mongoose.Schema({

name: {

type: String, // data validation here

required: true

},

email: {

type: String,

required: true

}

})

Create a model by passing in what I want to call the model and what schema to apply

const ContactModel = mongoose.model('contact', ContactSchema)

Export both the model and the schema

module.exports = {

ContactSchema,

ContactModel

}

No need to change the routes.js, form, index.js

Need to change controller because currently the contacts are stored in an array now we want to use mongodb instead.

On contact\_controller.js, bring in the model at the very top

const {

ContactModel

} = require('../models/contact\_model') // destructure becasue the export is an object

In the create function in contact\_controller.js. Add

ContactModel.create(contact).then(contact => {

res.render('layouts/main', {

view: 'success',

title: 'Thanks!'

})

}).catch(err => {

res.status(500).send(`Error: ${err}`)

})

Comment out the previous

// contacts.push(contact)

// console.log("contacts:", contacts)

Test app out on localhost:3000 and mongo console in the terminal

(mongo > show dbs > use contacts > db.contacts.find() = this should reveal that you have submitted a new contact in your app)

Add a show all contacts method in contact\_controller.js

let index = (req, res) => {

// return contacts promise because there's more than one

ContactModel.find().then(contacts => {

res.render('layouts/main', {

view: 'contacts',

title: 'All Contacts',

contacts

})

}).catch(err => {

res.status(500).send(`Error: ${err}`) // if promise rejects

})

}

Add to the export statement at the end of contact-Controller

module.exports = {

newContact,

create,

index

}

Add a new route to route.js

router.get('/contacts', index)

create a new view for showing all contacts in view > layouts called contacts.ejs

<h1>All Contacts</h1>

<ul>

<% for (contact of contacts) { %>

<li><b><%= contact.name %></b> (<%= contact.email %>)</li>

<% } %>

</ul>

Amend the below in main.ejs (make sure all the ejs views are sitting inside views > layouts)

<%- include("./" + view) -%>

To add a new field

in contact.ejs, add this field

<div>

<label>How can we help?</label>

<textarea name="enquiry"></textarea>

</div>

Update the model contact\_model

const ContactSchema = new mongoose.Schema({

name: {

type: String, // data validation here

required: true

},

email: {

type: String,

required: true

},

enquiry: {

type: String,

default: '' // even if they provide any data it will just be an empty string

}

})

Update the contact\_controller.js by adding the enquiry key

let create = (req, res) => {

// creates a contact object

let {

name,

email,

enquiry

} = req.body // takes req.body and extract a name and email property if exsits and createa name and email variable with the values of those properties (destructuring)

let contact = {

name,

email,

enquiry

} // create a new variable using the destructure data

In contacts.ejs add the following

<h1>All Contacts</h1>

<ul>

<% for (contact of contacts) { %>

<li><b><%= contact.name %></b> (<%= contact.email %>)</li>

<% if (contact.enquiry) { %>

<li style="margin-left: 2rem;">Enquiry: <%= contact.enquiry %></li>

<% } %>

<% } %>

</ul>