const {MongoClient, ObjectID} = require('mongodb')

//const MongoClient = mongodb.MongoClient

const url = 'mongodb://127.0.0.1:27017'

const databaseName = 'myDB';

MongoClient.connect(url, {

useNewUrlParser : true

}, (error, client) => {

if (error){

return console.log("Error connecting database!")

}

console.log("Database connected")

const db = client.db(databaseName)

const newID = ObjectID()

console.log("This is id ==>",newID)

db.collection('profiles').insert({

\_id:newID,

name: "najam",

email: "najam@gmail.com"

}, (error, response) => {

if (error){

return console.log("Error adding documents",error)

}

console.log(response.ops)

})

client.close()

})

Mongodb.js In this we create id Byself

const {MongoClient, ObjectID} = require('mongodb')

//const MongoClient = mongodb.MongoClient

const url = 'mongodb://127.0.0.1:27017'

const databaseName = 'myDB';

MongoClient.connect(url, {

useNewUrlParser : true

}, (error, client) => {

if (error){

return console.log("Error connecting database!")

}

console.log("Database connected")

const db = client.db(databaseName)

const newID = ObjectID()

console.log("This is id ==>",newID)

db.collection('profiles').insert({

\_id:newID,

name: "najam",

email: "najam@gmail.com"

}, (error, response) => {

if (error){

return console.log("Error adding documents",error)

}

console.log(response.ops)

})

client.close()

})

Mongodb.js: Through Find to get Onedata

const {MongoClient, ObjectID} = require('mongodb')

//const MongoClient = mongodb.MongoClient

const url = 'mongodb://127.0.0.1:27017'

const databaseName = 'myDB';

MongoClient.connect(url, {

useNewUrlParser : true

}, (error, client) => {

if (error){

return console.log("Error connecting database!")

}

console.log("Database connected")

const db = client.db(databaseName)

//const newID = ObjectID()

//console.log("This is id ==>",newID)

db.collection('profiles').findOneAndDelete({

//\_id:newID,

//name: "najam",

//email: "najam@gmail.com"

email:'ahmad.ishtiaq990@gmail.com'

}, (error, data) => {

if (error){

return console.log("Error adding documents",error)

}

console.log(data)

})

client.close()

})

Mongodb.js : To give ID and get the result

const {MongoClient, ObjectID} = require('mongodb')

//const MongoClient = mongodb.MongoClient

const url = 'mongodb://127.0.0.1:27017'

const databaseName = 'myDB';

MongoClient.connect(url, {

useNewUrlParser : true

}, (error, client) => {

if (error){

return console.log("Error connecting database!")

}

console.log("Database connected")

const db = client.db(databaseName)

//const newID = ObjectID()

//console.log("This is id ==>",newID)

db.collection('profiles').findOneAndDelete({

//\_id:newID,

//name: "najam",

//email: "najam@gmail.com"

\_id:ObjectID("5d09fa7348f9282874271e73")

}, (error, data) => {

if (error){

return console.log("Error adding documents",error)

}

console.log(data)

})

client.close()

})

Mongo.db: Through find

const {MongoClient, ObjectID} = require('mongodb')

//const MongoClient = mongodb.MongoClient

const url = 'mongodb://127.0.0.1:27017'

const databaseName = 'myDB';

MongoClient.connect(url, {

useNewUrlParser : true

}, (error, client) => {

if (error){

return console.log("Error connecting database!")

}

console.log("Database connected!!")

const db = client.db(databaseName)

db.collection('profiles').find({

email: 'ahmed.ishtiaq990@gmail.com'

}).count((error, count) => console.log(count))

//const newID = ObjectId()

//console.log(newID)

//db.collection('profiles').find({

// \_id:ObjectID("5d09fa7348f9282874271e73")

// }) .toArray ((error, arrayData)) => {

client.close()

})

Mongodb.js: To get array

const {MongoClient, ObjectID} = require('mongodb')

//const MongoClient = mongodb.MongoClient

const url = 'mongodb://127.0.0.1:27017'

const databaseName = 'myDB';

MongoClient.connect(url, {

useNewUrlParser : true

}, (error, client) => {

if (error){

return console.log("Error connecting database!")

}

console.log("Database connected!!")

const db = client.db(databaseName)

db.collection('profiles').find({

email: 'ahmed.ishtiaq990@gmail.com'

}).toArray((error, dataArray) => {

console.log(dataArray)

})

//const newID = ObjectId()

//console.log(newID)

//db.collection('profiles').find({

// \_id:ObjectID("5d09fa7348f9282874271e73")

// }) .toArray ((error, arrayData)) => {

client.close()

})

Mongodb.js When we use $set

const {MongoClient, ObjectID} = require('mongodb')

//const MongoClient = mongodb.MongoClient

const url = 'mongodb://127.0.0.1:27017'

const databaseName = 'myDB';

MongoClient.connect(url, {

useNewUrlParser : true

}, (error, client) => {

if (error){

return console.log("Error connecting database!")

}

console.log("Database connected!!")

const db = client.db(databaseName)

db.collection('profiles').updateOne(

{email: 'ahmad.ishtiaq990@gmail.com'},

{$set: {name: "Ish Siddiqui" }}

).then((data) => {

console.log(data)

}).catch(e => console.log("Error updating document"))

//db.collection('profiles').find({

//email: 'ahmed.ishtiaq990@gmail.com'

//}).toArray((error, dataArray) => {

//console.log(dataArray)

//})

//const newID = ObjectId()

//console.log(newID)

//db.collection('profiles').find({

// \_id:ObjectID("5d09fa7348f9282874271e73")

// }) .toArray ((error, arrayData)) => {

client.close()

})

Mongodb.js : To use $inc

const {MongoClient, ObjectID} = require('mongodb')

//const MongoClient = mongodb.MongoClient

const url = 'mongodb://127.0.0.1:27017'

const databaseName = 'myDB';

MongoClient.connect(url, {

useNewUrlParser : true

}, (error, client) => {

if (error){

return console.log("Error connecting database!")

}

console.log("Database connected!!")

const db = client.db(databaseName)

db.collection('profiles').updateOne(

{email: 'ahmad.ishtiaq990@gmail.com'},

{$inc: {age: 1 }}

).then((data) => {

console.log(data)

}).catch(e => console.log("Error updating document"))

//db.collection('profiles').find({

//email: 'ahmed.ishtiaq990@gmail.com'

//}).toArray((error, dataArray) => {

//console.log(dataArray)

//})

//const newID = ObjectId()

//console.log(newID)

//db.collection('profiles').find({

// \_id:ObjectID("5d09fa7348f9282874271e73")

// }) .toArray ((error, arrayData)) => {

client.close()

})

**Index.js**

require('./db/mongoose')

const Profiles = require('./models/profiles')

const newRec = Profiles({

name: "Ishtiaq",

age: 74,

graduate: true,

email: "ahmad.ishtiaq990@gmail.com"

})

newRec.save()

.then(data => console.log(data))

.catch(err => console.log(err))

**Profiles.js**

const mongoose = require('mongoose')

const Profiles = mongoose.model('Profiles', {

name: {

type: String

},

age: {

type: Number

},

graduate: {

type: Boolean

},

email: {

type: String

}

})

module.exports = Profiles

**mongoose.js**

const mongoose = require ('mongoose')

mongoose.connect('mongodb://127.0.0.1:27017/mydb-mongoose',{

useNewUrlParser: true,

useCreateIndex: true

})

**Package.json**

{

"name": "MONGO-DB",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

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

},

"keywords": [],

"author": "",

"license": "ISC",

"dependencies": {

"express": "^4.17.1",

"mongodb": "^3.2.7",

"mongoose": "^5.6.0"

}

}

These coding’s are used to use POST

**Index.js**

const express = require ('express')

require('./db/mongoose')

const Profiles = require('./models/profiles')

// newRec.save()

// .then(data => console.log(data))

// .catch(err => console.log(err))

const app = express()

const port = process.env. PORT || 3000

app.use(express.json())

app.post('/profiles', (req, res) => {

const profile = Profiles(req.body)

profile.save().then (() => {

res.send(profile)

}).catch((e) => {

res.status(400)

res.send(e)

})

})

**Profiles.js**

const mongoose = require('mongoose')

const validator = require('validator')

const Profiles = mongoose.model('Profiles', {

name: {

type: String

},

age: {

type: Number,

min: 0,

validate (value) {

if (value < 0){

throw new Error ("Age cannot be negative")

}

}

},

graduate: {

type: Boolean,

required: true,

default: false

},

email: {

type: String,

lowercase: true,

validate(value) {

if (!validator.isEmail(value)) {

throw new Error ("Expevted correct email")

}

}

}

})

module.exports = Profiles

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZzzzzzzzzzzzzzzzzz