Node

**Npm init -y**

**Npm i express nodemon mongoose**

**”start” : ”nodemon server.js”**

**Now npm start**

**Server.js File**

const express = require('express')

const app = express()

const mongoose = require('mongoose')

const PORT = 5000;

const {MONGOURI} = require('./keys');

mongoose.connect(MONGOURI,{

useNewUrlParser:true,

useUnifiedTopology: true

})

mongoose.connection.on('connected',()=>{

console.log("conneted to mongo yeahh")

})

mongoose.connection.on('error',(err)=>{

console.log("err connecting",err)

})

require('./models/user')

app.use(express.json())

app.use(require('./routes/auth'))

app.listen(PORT,()=>{

console.log("server is running on",PORT)

})

**Keys.js**

module.exports={

MONGOURI: "mongodb+srv://admin:123@cluster0.fdok7.mongodb.net/Cluster0?retryWrites=true&w=majority"

}

Folder **models/user.js**

const mongoose = require('mongoose')

const userSchema = new mongoose.Schema({

name:{

type:String,

required:true

},

email:{

type:String,

required:true

},

password:{

type:String,

required:true

}

})

mongoose.model("User",userSchema)

Folder **routes/auth.js**

const express = require('express')

const router = express.Router()

const mongoose = require('mongoose')

const User = mongoose.model("User")

router.post('/signup',(req,res)=>{

const {name,email,password} = req.body

if(!email || !password || !name){

return res.status(422).json({error:"please add all the fields"})

}

User.findOne({email:email})

.then((savedUser)=>{

if(savedUser){

return res.status(422).json({error:"user already exists with that email"})

}

const user = User({

name,

email,

password

})

user.save()

.then(user=>{

res.json({message: 'Saved Successfully'});

})

.catch(err=>{

console.log(err);

})

})

.catch(err=>{

console.log(err);

})

})

module.exports = router

Now we go to postman and try to add user command+shift+4

A screenshot of a social media post

Description automatically generated

**Hashing Password**

**We need to install** npm install bcryptjs

After we change our auth.js file

We add this const bcrypt = require('bcryptjs');

And saved code insert into the bcrypt function

bcrypt.hash(password,12).then(hashedpassword =>{

const user = User({

name,

email,

password:hashedpassword

})

user.save()

.then(user=>{

res.json({message: 'Saved Successfully'});

})

.catch(err=>{

console.log(err);

})

})

Now we create the signin route to auth.js file

router.post('/signin',(req,res)=>{

const {email,password} = req.body

if(!email || !password){

return res.status(422).json({error:"please add email or password"})

}

User.findOne({email:email})

.then(savedUser=>{

if(!savedUser){

return res.status(422).json({error:"Invalid Email or password"})

}

bcrypt.compare(password,savedUser.password)

.then(doMatch=>{

if(doMatch){

res.json({message:"successfully signed in"})

}

else{

return res.status(422).json({error:"Invalid Email or password"})

}

})

.catch(err=>{

console.log(err)

})

})

})

So we test to postman

A screenshot of a social media post

Description automatically generated

**JWT ( json web token )**

**Install** npm i jsonwebtoken

Add this line to auth.js file

const jwt = require('jsonwebtoken');

now we need a scret key to sign in the token

so we goto our key.js file and add this line

module.exports={

MONGOURI: "mongodb+srv://admin:123@cluster0.fdok7.mongodb.net/Cluster0?retryWrites=true&w=majority",

JWT\_SECRET:"dgfgdsfgdsfgcdsghfcfdg"

}

Add this line to auth.js file

const {JWT\_SECRET}= require('../keys');

//create a token and send as response

if(doMatch){

*//res.json({message:"successfully signed in"})*

const token = jwt.sign({\_id:savedUser.\_id},JWT\_SECRET);

res.json({token});

}

So if we run this application we can see

A screenshot of a social media post

Description automatically generated

**Now create a middleware to protected the route only login user can access information**

**So we create a folder called middleware and inside we create file called requireLogin.js** we edit the file like this

const jwt = require('jsonwebtoken')

const {JWT\_SECRET} = require('../keys')

const mongoose = require('mongoose')

const User = mongoose.model("User")

module.exports = (req,res,next)=>{

const {authorization} = req.headers

*//authorization === Bearer ewefwegwrherhe*

if(!authorization){

return res.status(401).json({error:"you must be logged in"})

}

const token = authorization.replace("Bearer ","")

jwt.verify(token,JWT\_SECRET,(err,payload)=>{

if(err){

return res.status(401).json({error:"you must be logged in"})

}

const {\_id} = payload

User.findById(\_id).then(userdata=>{

req.user = userdata

next()

})

})

}