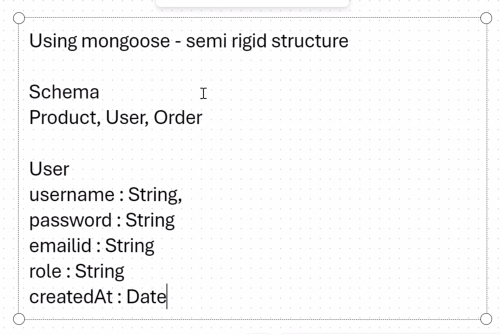
mongodb+srv://<username>:<password>@<clusteraddress>/<dbname>?retrywrites=true&w=majority

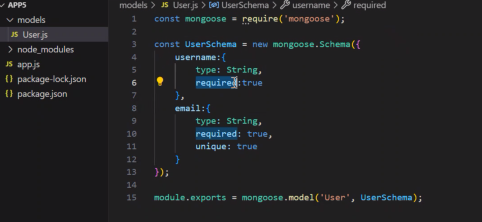
connecting to mongodb atlas

mongoose.connect(mongoURI)

    .then(()=> console.log('Mongodb connected.....'))

    .catch(err=> console.log(err));





const mongoose = require('mongoose');

const UserSchema = new mongoose.Schema({

    username:{

        type: String,

        required:true

    },

    email:{

        type: String,

        required: true,

        unique: true

    },

    password:{

        type: String,

        required: true,

        minLength: 8

    }

});

module.exports = mongoose.model('User', UserSchema);

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

    res.send('Connected to MongoDB');

});

app.get('/api/users', async (req,res)=>{

    try{

        const users = await User.find();

        res.json(users);

    } catch(err){

        res.status(500).send(err);

    }

});

app.post('/api/users',async (req,res)=>{

    try{

        const userData = req.body;

        const user = await User.create(userData);

        res.status(201).json(user);

    } catch(err){

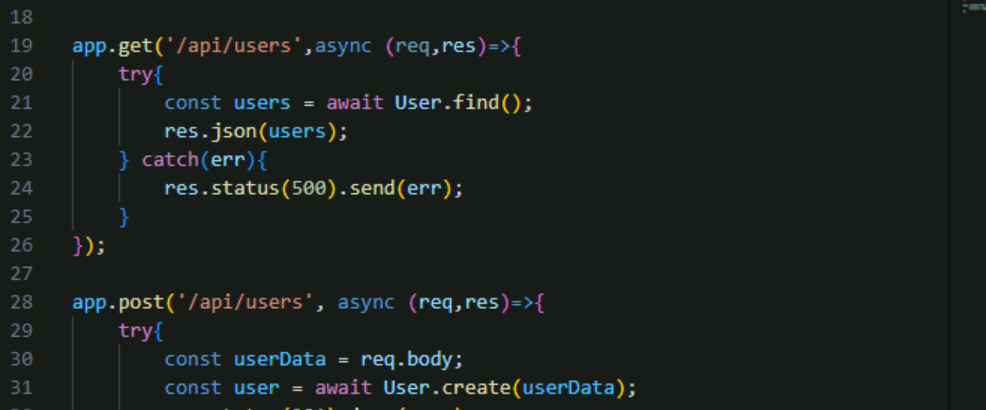
        res.status(500).send(err);

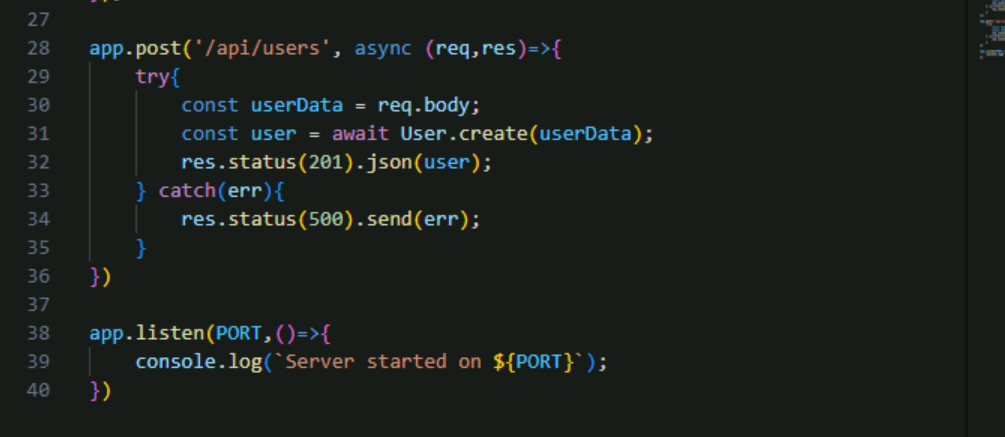
    }

})

App.js







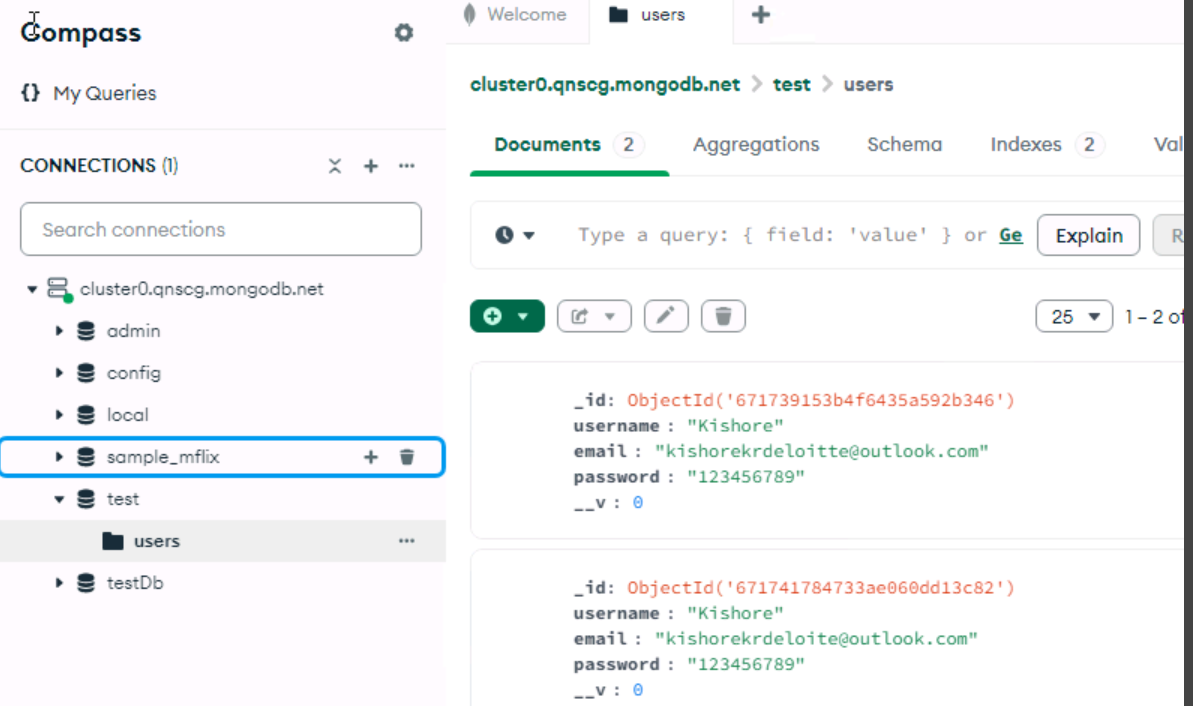
User.js

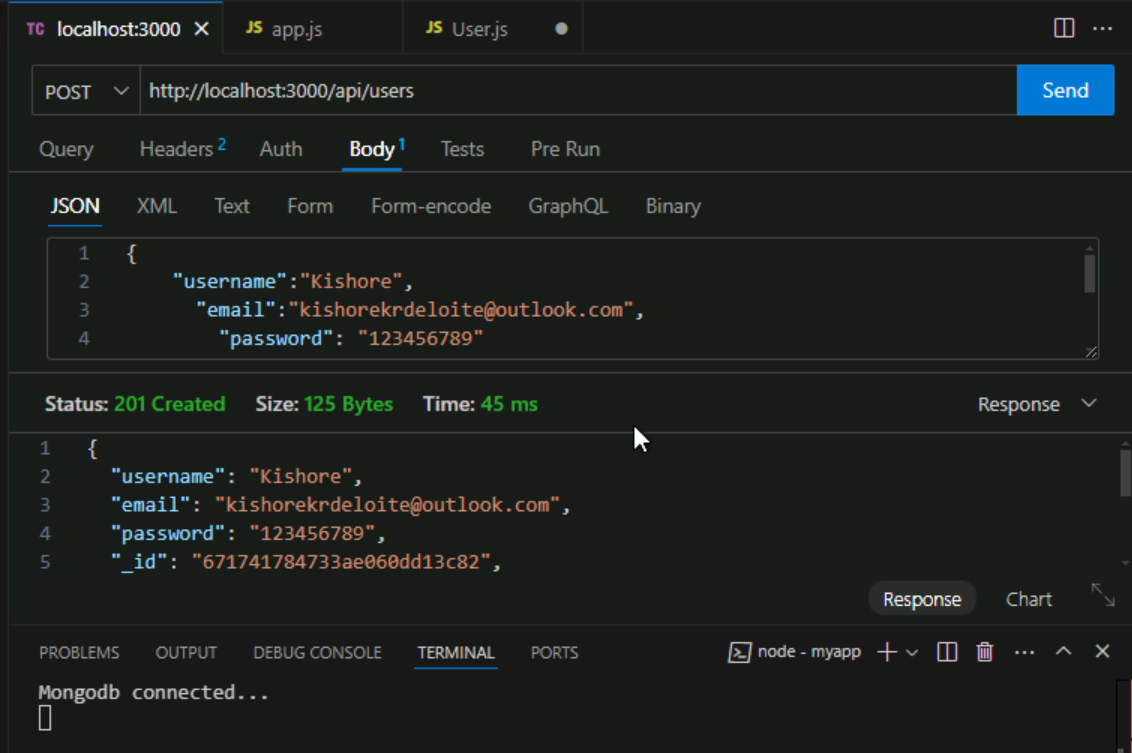


Storing password in this way is not secure so we must encrypt it

To encrypt the password first install a module

Npm install bcrypt



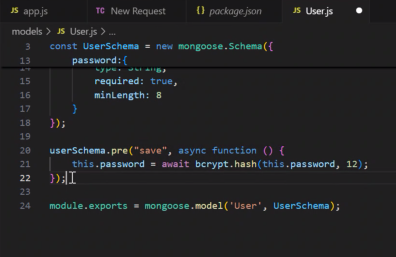


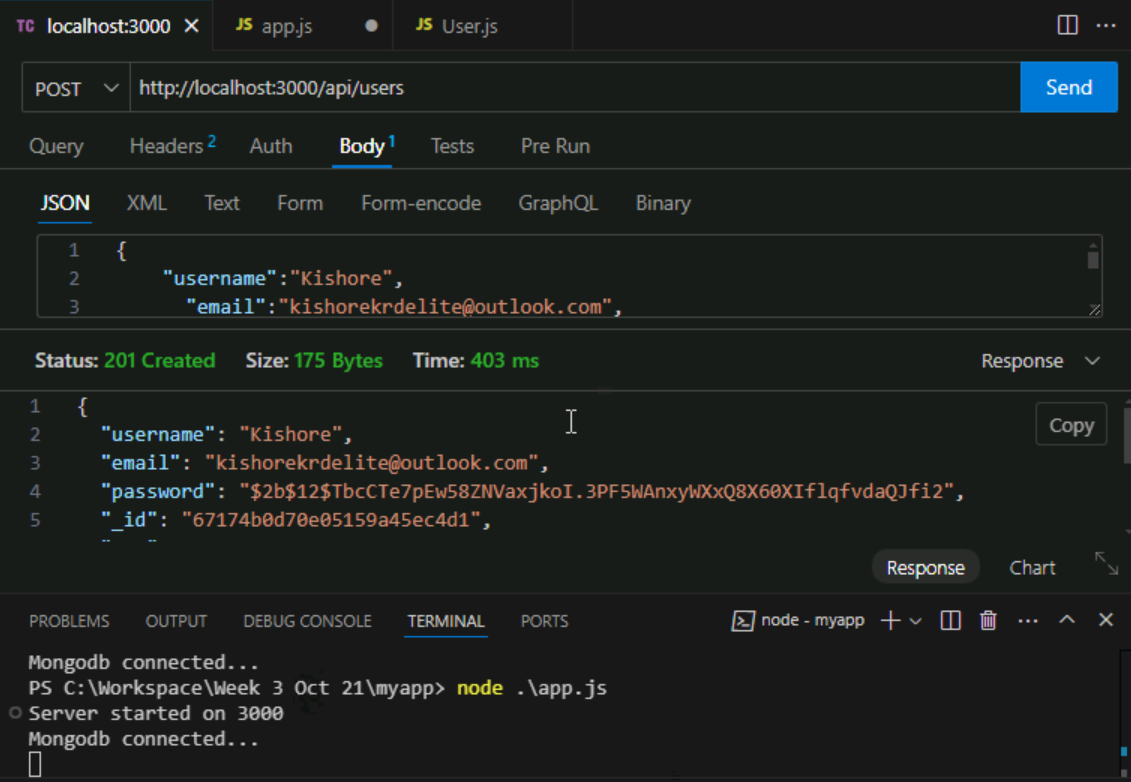
Storing password in this way is not secure so we must encrypt it

To encrypt the password first install a module

Npm install bcrypt

Add $ const bcrypt = require(‘bcrypt’) to import the module



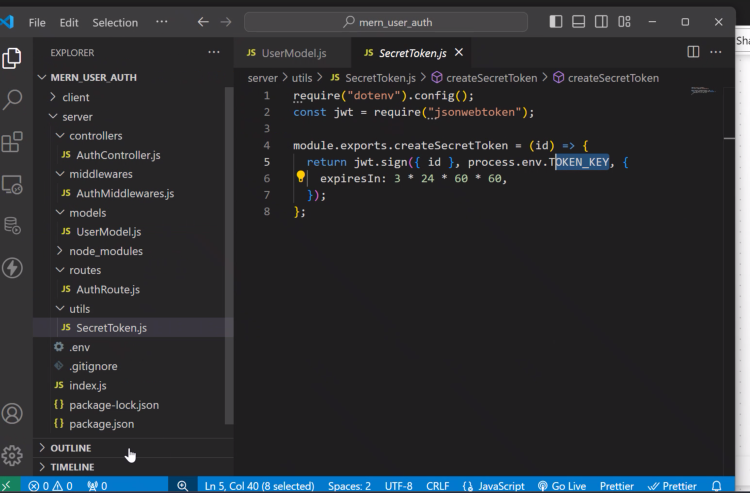


Now the password is encrypted into a hash value

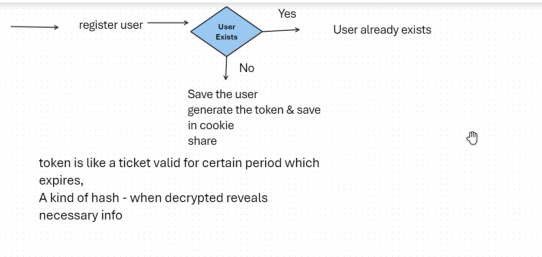
Github link viswas (backend app which authenticates user auth)

<https://github.com/vishwasks32/mernappbackend.git>

Token in viswas example code//



User comes in and tried to login in



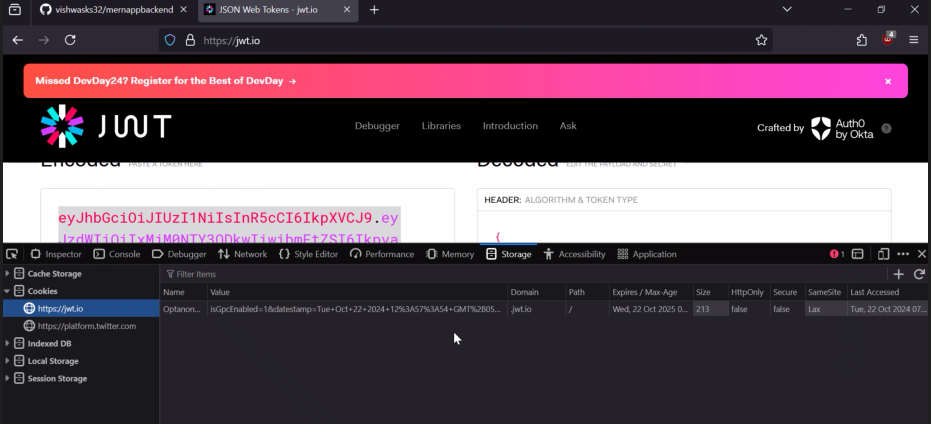


JSON Web Token link

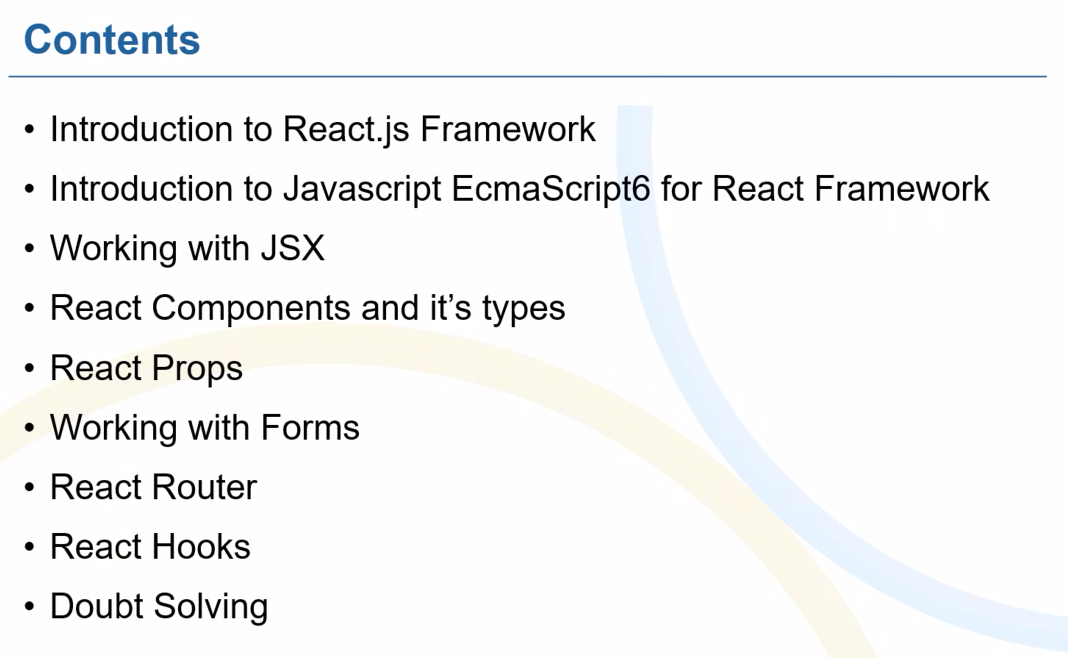
<https://jwt.io/>

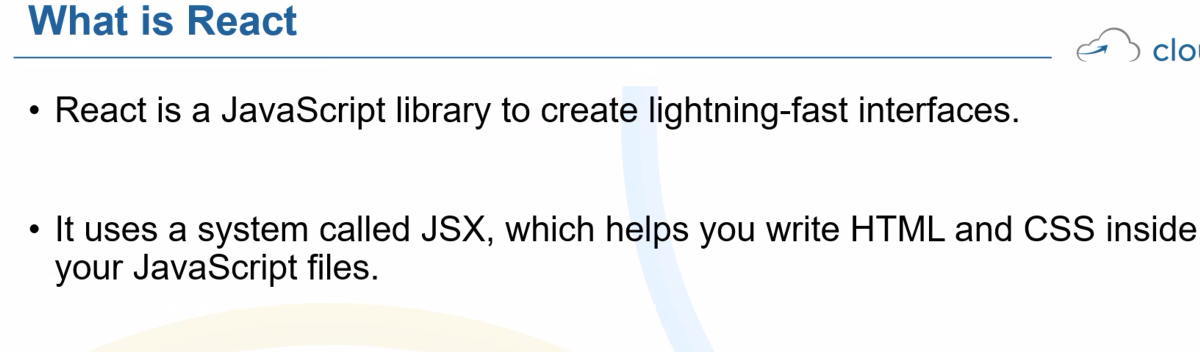
First thing – emcrypting the passs and saving it.

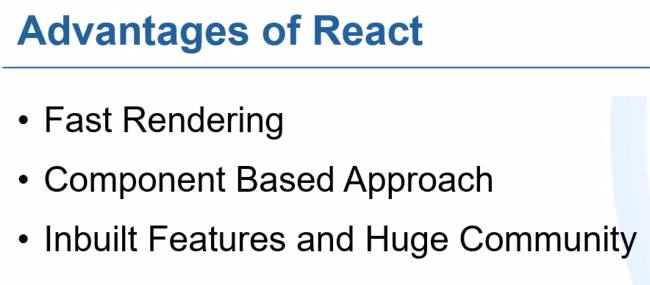
Second thing – not using the pass but using jwt token which shows it in cncrypted form and deletes after a particular amount of time. So where to store it, can be stored in side a cookie and we can store the token and whenever the user visits if the token has not expired the user can login or else have to login again. Cookies would basically ve name and a value.

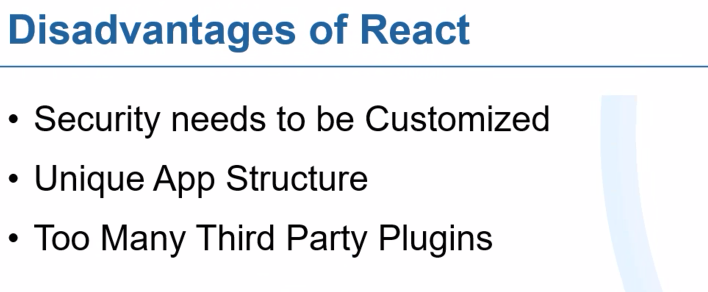


But recently people are moving from cookies storage to session storage. Which is more secure.





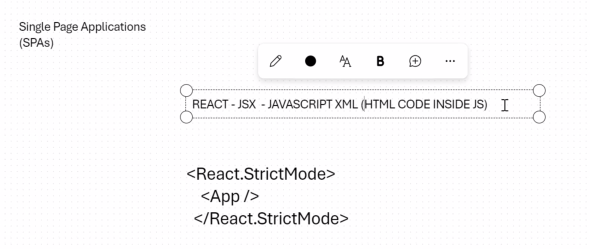


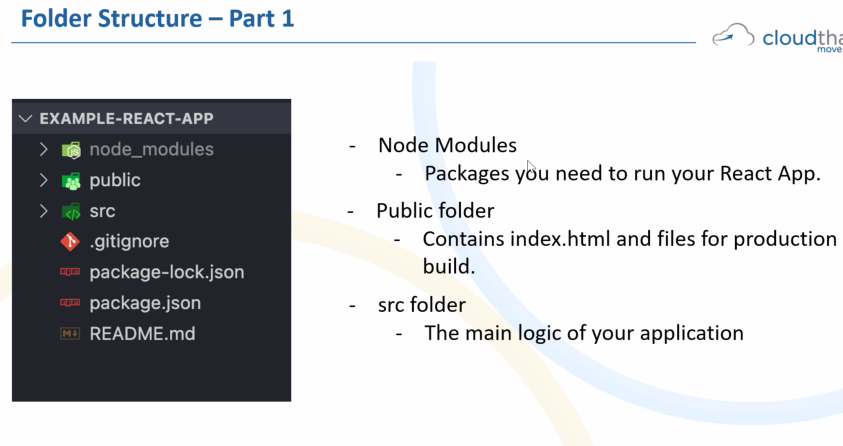


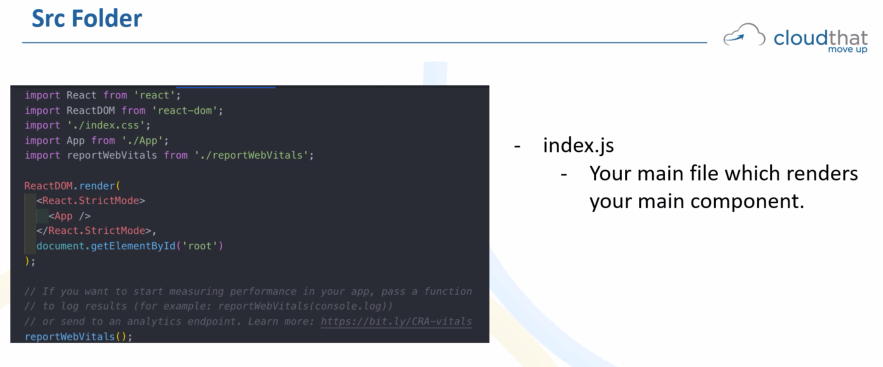
To create a react app:

Create a new file>terminal>

npx create-react-app app1





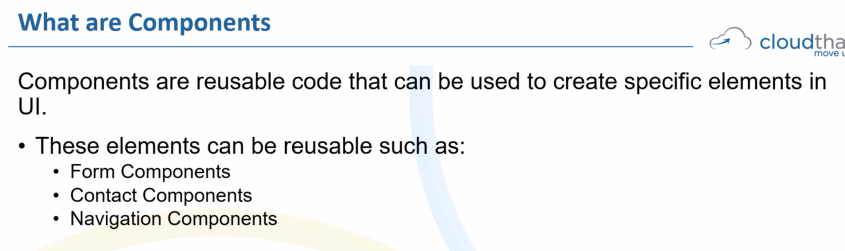


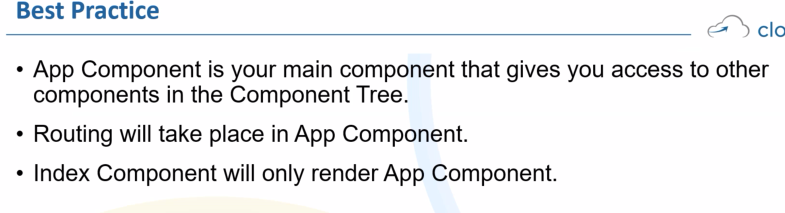
Basic building block of a react app is a component.

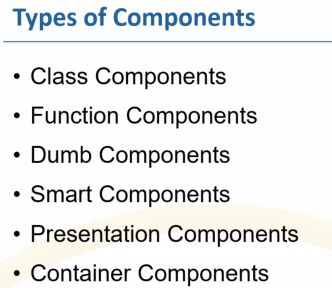
Two types of Components:

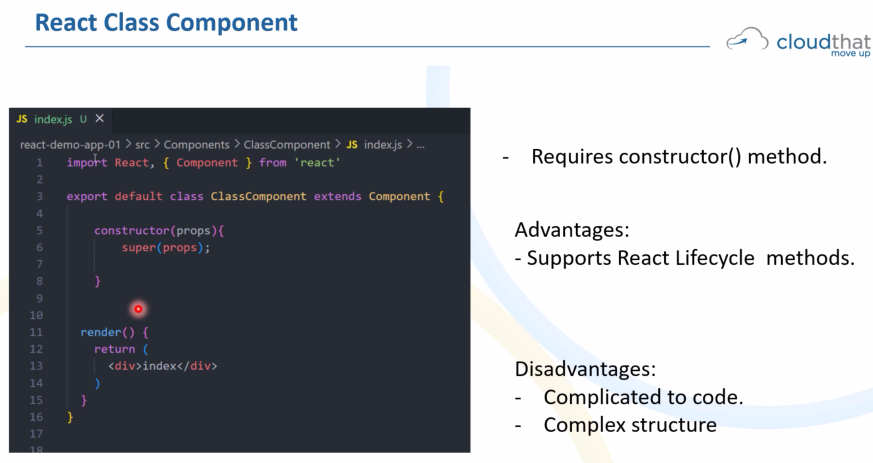
1.Functional Component – basically a JS Function which renders a HTML – recent versions of react use this more from (16.8) version.

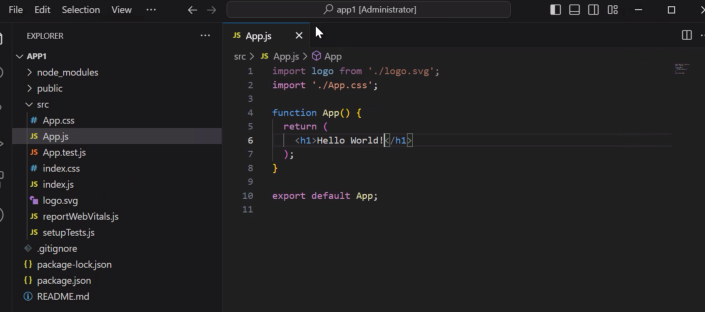
2.Class Component. – basically a JS-ES6 Classes, State, lifecycle methods.



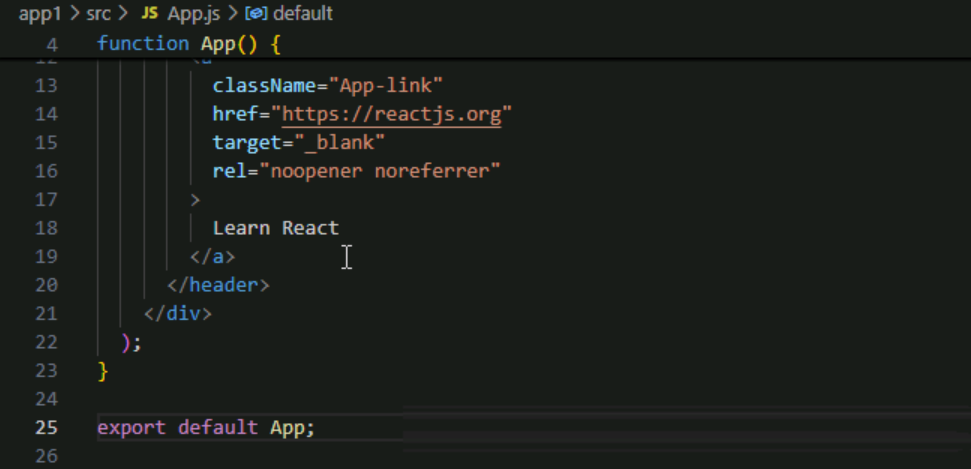




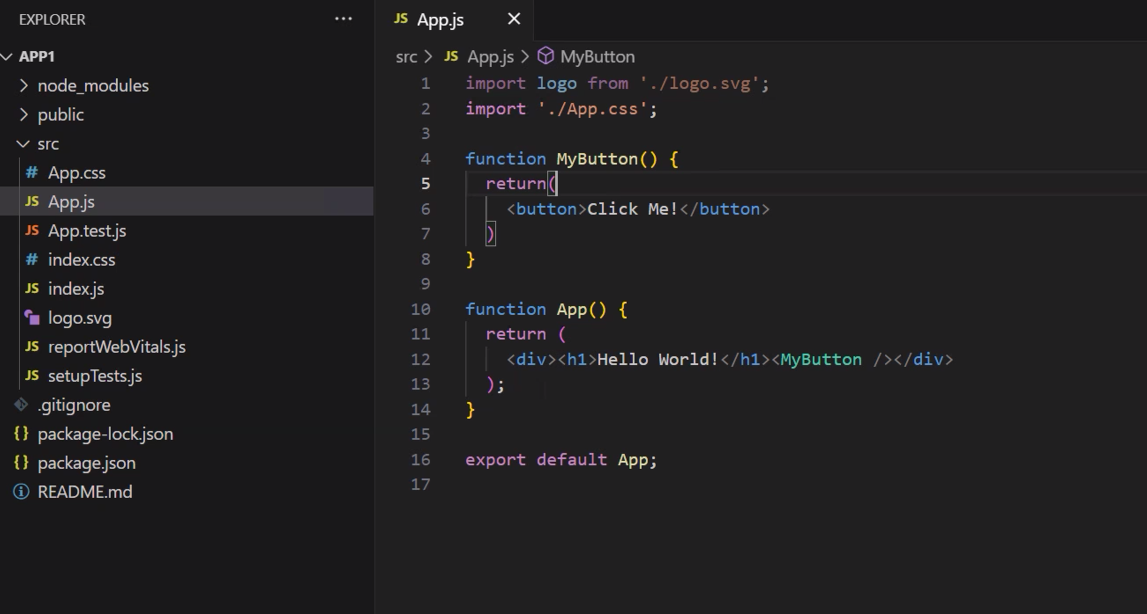


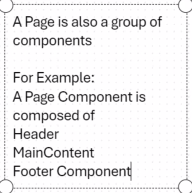


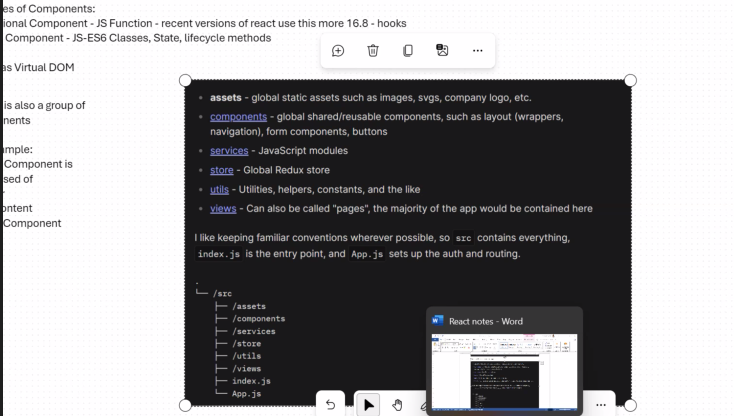


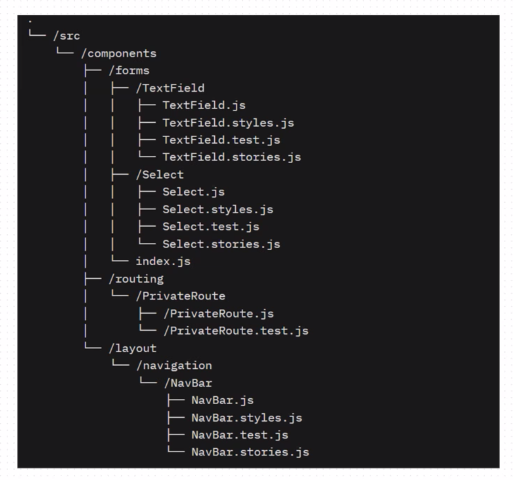


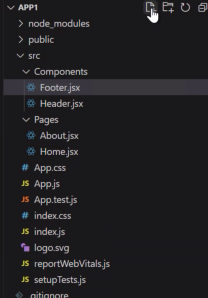
React has virtual dom



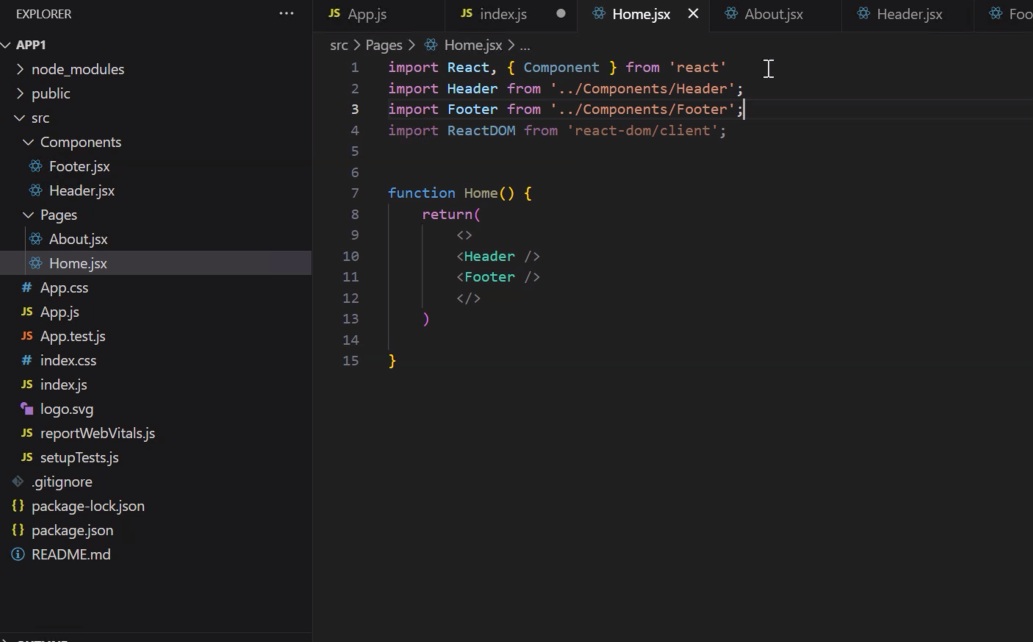


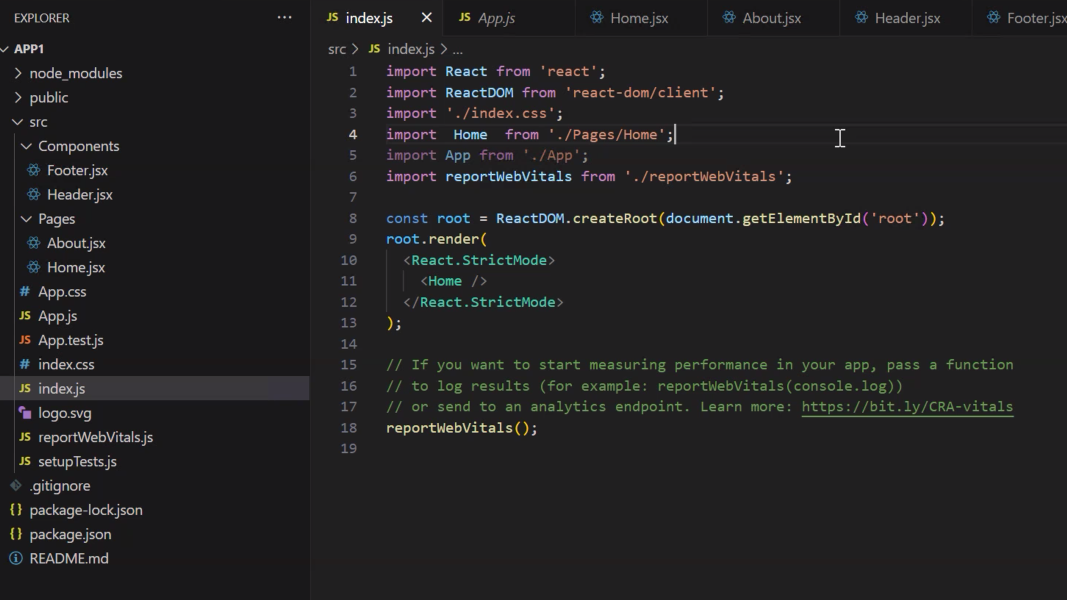






What is JSX fragment?





import React, { Component } from 'react'

function Header() {

    return (

        <h1>My App</h1>

    )

}

export default Header;

(Header.jsx)

Learn react:

https://react.dev/learn