**Title: Demonstration of component intercommunication using ReactJS**

npx create-react-app my-app

cd my-app

npm start

Parentcomponent.js

import React, { useState } from 'react';

import ChildComponent from './ChildComponent';

const ParentComponent = () => {

const [message, setMessage] = useState('');

const handleMessageChange = (newMessage) => { setMessage(newMessage);

};

return (

<div>

<h2>Parent Component</h2>

<ChildComponent message={message} onMessageChange={handleMessageChange} />

<p>Message from ChildComponent: {message}</p>

</div>

);

};

export default ParentComponent;

ChildComponent.js

import React from 'react';

const ChildComponent = ({ message, onMessageChange }) => {

  // Handle the Enter key press

  const handleKeyDown = (event) => {

    if (event.key === 'Enter') { // Check if the Enter key is pressed

      const newMessage = event.target.value;

      onMessageChange(newMessage); // Call the parent's function

    }

  };

  return (

    <div>

      <h3>Child Component</h3>

      <input

        type="text"

        defaultValue={message} // Use defaultValue since value binding isn't necessary here

        onKeyDown={handleKeyDown} // Trigger on Enter key press

      />

    </div>

  );

};

export default ChildComponent;

app.js

// import logo from './logo.svg';

import './App.css';

import React from 'react';

import ParentComponent from './ParentComponent.js';

// import ChildComponent from './ChildComponent.js';

function App() {

  return (

    <div><ParentComponent /></div>

  );

}

export default App;

**Title: Demonstration of authentication and authorization using Express.**

Views/form.ejs

<html>

    <body>

        <form action="/login" method="post">

            <input type="text" placeholder="username" name="username">

            <input type="password" placeholder="password" name="password">

            <button type="submit">submit</button>

        </form>

    </body>

</html>

Index.js

const express = require('express');

const bcrypt = require('bcrypt');

const jwt = require('jsonwebtoken');

const path = require('path');

const app = express();

const PORT = 3000;

app.use(express.json());

app.use(express.urlencoded({ extended: true }));

app.set('view engine', 'ejs');

app.set('views', path.join(\_\_dirname, 'views'));

const users = [

{ id: 1, username: 'user1', password: bcrypt.hashSync('password1', 10), role: 'user' },

{ id: 2, username: 'admin', password: bcrypt.hashSync('password2', 10), role: 'admin' }

];

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

    res.render('form')

})

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

const { username, password } = req.body;

console.log(username)

console.log(password)

const user = users.find((user) => user.username === username);

if (!user) {

return res.status(401).json({ message: 'Invalid username' });

}

if (!bcrypt.compareSync(password, user.password)) {

return res.status(401).json({ message: 'Invalid username' });

}

const token = jwt.sign({ id: user.id, username: user.username, role: user.role }, 'secretkey');

res.json({ token });

// res.json({ msg:"welcome" });

});

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

res.json({ message: 'Protected route accessed successfully' });

});

function authenticateToken(req, res, next) {

const authHeader = req.headers['authorization'];

const token = authHeader && authHeader.split(' ')[1];

if (!token) {

return res.sendStatus(401);

}

jwt.verify(token, 'secretkey', (err, user) => {

if (err) {

return res.sendStatus(403);

}

req.user = user;

next();

});

}

app.listen(PORT, () => {

console.log(`Server is running on port ${PORT}`);

});

**Title: To Access Data Using Nodejs**

File.js

const mongoose = require("mongoose")

const url='mongodb://localhost:27017/mydb'

const connectDB = async ()=>{

 try {

 await mongoose.connect(url)

 console.log('connected to DB');

 } catch (error) {

 console.log(error);

 }

}

const Userschema = new mongoose.Schema({

 name:String,

 age:Number,

})

const Model=mongoose.model("Model",Userschema)

console.log(Model);

connectDB();

const userCreate=async()=>{

 const user=await Model.create({name:'Person1',age:20})

 console.log(user)

 console.log('created');

}

// userCreate()

const userfind=async()=>{

    const user=await Model.find();

    console.log(user)

    console.log('Find completed');}

// userfind()

const userfindbyid=async()=>{

    const user=await Model.findById('67433f8a8269fa3026e3b651') ;

    console.log(user)

    console.log('Findbyid Completed');}

userfindbyid()

**Title: To Create a form to edit the data using Angular2.**

npm install -g @angular/cli

ng new myNewApp

cd myNewApp

ng serve

app.component.html

<div style="max-width: 300px; margin: auto;">

  <h2>Edit User Information</h2>

  <form [formGroup]="editDataForm" (ngSubmit)="onSubmit()">

    <div>

      <label for="name">Name</label>

      <input id="name" type="text" formControlName="name" />

    </div>

    <div>

      <label for="email">Email</label>

      <input id="email" type="email" formControlName="email" />

    </div>

    <button type="submit">Submit</button>

  </form>

  <!-- Displaying edited user data -->

  <div \*ngIf="submittedUser">

    <h3>Submitted Data:</h3>

    <p><strong>Name:</strong> {{ submittedUser.name }}</p>

    <p><strong>Email:</strong> {{ submittedUser.email }}</p>

  </div>

</div>

app.component.ts

import { Component } from '@angular/core';

import { ReactiveFormsModule, FormGroup, FormControl } from '@angular/forms';

import { CommonModule } from '@angular/common';

@Component({

  selector: 'app-root',

  standalone: true,

  imports: [CommonModule, ReactiveFormsModule],

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  // Define the form group

  editDataForm: FormGroup;

  // Variable to hold submitted data

  submittedUser: { name: string, email: string } | null = null;

  constructor() {

    // Initialize form with default values

    this.editDataForm = new FormGroup({

      name: new FormControl('John Doe'),

      email: new FormControl('john.doe@example.com'),

    });

  }

  // Function to handle form submission

  onSubmit() {

    // Assign form values to submittedUser

    this.submittedUser = this.editDataForm.value;

    // Optionally, log the data to the console

    console.log('Form Data:', this.submittedUser);

  }

}

main.ts file

import { bootstrapApplication } from '@angular/platform-browser';

import { AppComponent } from './app/app.component';

bootstrapApplication(AppComponent)

 .catch(err => console.error(err));