import { Component, Input, Output, EventEmitter } from '@angular/core';

import { LogicService } from '../../../services/logic.service';

import { ConstantsService } from '../../../services/constants.service';

@Component({

selector: 'app-side-menu',

templateUrl: './side-menu.component.html',

styleUrls: ['./side-menu.component.css']

})

export class SideMenuComponent {

@Input() project;

@Input() criteriaPairwise = [];

@Input() alternativesPairwise = [];

@Output() notifyNodeChanges = new EventEmitter<any>();

@Output() notifyCalculateDecision = new EventEmitter<any>();

@Output() notifySaveProject = new EventEmitter<any>();

evaluators = [];

constructor(private logicService: LogicService,

private constantsService: ConstantsService) { }

addAlternative() {

if(this.project.imMaster) {

let index = this.project.alternatives.length+1;

this.project.alternatives.push({ name:'Alternativa # ' + index, description:''});

for (let i = 0; i < this.project.criteria.length; i++) {

this.project.alternativesComparisons[i] = this.logicService.addComparisons(this.project.alternativesComparisons[i], this.project.alternatives.length);

}

this.notifyNodeChanges.emit();

}

}

removeAlternative (alternative) {

if(this.project.imMaster) {

let indexAlternative = this.project.alternatives.indexOf(alternative)

this.project.alternatives.splice(this.project.alternatives.indexOf(alternative),1);

for (let i = 0; i < this.project.criteria.length; i++) {

this.project.alternativesComparisons[i] = this.logicService.removeComparisons(

indexAlternative, this.project.alternativesComparisons[i], this.project.alternatives.length);

}

this.notifyNodeChanges.emit();

}

}

addCriterion() {

if(this.project.imMaster) {

let index= this.project.criteria.length+1;

this.project.criteria.push({ name:'Criterio # ' + index, description:''});

this.project.alternativesComparisons.push([]);

for (let j = 0; j < this.alternativesPairwise .length; j++) {

this.project.alternativesComparisons[index - 1].push(this.constantsService.getSliderInitialvalue());

}

this.project.criteriaComparisons = this.logicService.addComparisons(this.project.criteriaComparisons, this.project.criteria.length);

this.notifyNodeChanges.emit();

}

}

removeCriterion (criterion) {

if(this.project.imMaster) {

let indexCriterion = this.project.criteria.indexOf(criterion);

this.project.criteria.splice(indexCriterion,1);

this.project.alternativesComparisons.splice(indexCriterion,1);

this.project.criteriaComparisons = this.logicService.removeComparisons(indexCriterion, this.project.criteriaComparisons, this.project.criteria.length);

this.notifyNodeChanges.emit();

}

}

addEvaluator() {

this.project.evaluators.push({email: 'evaluador'+(this.project.evaluators.length+1)+'@mail.com'});

}

removeEvaluator(evaluator) {

const indexEvaluator = this.project.evaluators.indexOf(evaluator);

this.project.evaluators.splice(indexEvaluator,1);

}

onCalcularClick() {

this.notifyCalculateDecision.emit();

}

onSaveClick() {

this.notifySaveProject.emit();

}

}