import { LightningElement, track } from 'lwc';

export default class TopicExplorer extends LightningElement {

@track searchTerm = '';

@track selectedTopicIds = new Set();

topics = [

{ id: '1', name: 'LWC Basics', description: 'Learn Lightning Web Components', link: 'https://example.com/lwc-basics' },

{ id: '2', name: 'Apex Programming', description: 'Server-side logic', link: 'https://example.com/apex' },

{ id: '3', name: 'Salesforce AI', description: 'Integrate AI in Salesforce', link: 'https://example.com/ai' },

];

get filteredTopics() {

return this.topics

.filter(topic => topic.name.toLowerCase().includes(this.searchTerm.toLowerCase()))

.map(topic => ({

...topic,

selectLabel: this.selectedTopicIds.has(topic.id) ? 'Deselect' : 'Select',

selectVariant: this.selectedTopicIds.has(topic.id) ? 'destructive' : 'neutral'

}));

}

handleSearch(event) {

this.searchTerm = event.target.value;

}

toggleSelect(event) {

const topicId = event.target.dataset.id;

if (this.selectedTopicIds.has(topicId)) {

this.selectedTopicIds.delete(topicId);

} else {

this.selectedTopicIds.add(topicId);

}

// Trigger reactive update

this.selectedTopicIds = new Set(this.selectedTopicIds);

}

navigateToLink(event) {

const url = event.target.dataset.id;

window.open(url, '\_blank');

}

get selectedTopics() {

return this.topics.filter(topic => this.selectedTopicIds.has(topic.id));

}

}