import { Lokacija } from "./lokacija.js";

export class Vrt {

    // Dodat je ID, da bi povezivanje sa bazom bilo lakše...

    constructor(id, naziv, n, m, kapacitetLok) {

        this.id = id;

        this.naziv = naziv;

        this.n = n;

        this.m = m;

        this.kapacitet = kapacitetLok;

        this.kontejner = null;

        this.lokacije = [];

    }

    dodajLokaciju(lok) {

        this.lokacije.push(lok);

    }

    crtajVrt(host) {

        if (!host)

            throw new Exception("Roditeljski element ne postoji");

        this.kontejner = document.createElement("div");

        this.kontejner.classList.add("kontejner");

        host.appendChild(this.kontejner);

        this.crtajFormu(this.kontejner);

        this.crtajLokacije(this.kontejner);

    }

    crtajFormu(host) {

        const kontForma = document.createElement("div");

        kontForma.className = "kontForma";

        host.appendChild(kontForma);

        var elLabela = document.createElement("h3");

        elLabela.innerHTML = "Unos zivotinja";

        kontForma.appendChild(elLabela);

        elLabela = document.createElement("label");

        elLabela.innerHTML = "Ime vrste";

        kontForma.appendChild(elLabela);

        let tb = document.createElement("input");

        tb.className = "vrsta";

        kontForma.appendChild(tb);

        elLabela = document.createElement("label");

        elLabela.innerHTML = "Kolicina";

        kontForma.appendChild(elLabela);

        tb = document.createElement("input");

        tb.className = "kolicina";

        tb.type = "number";

        kontForma.appendChild(tb);

        let tipoviStanista = ["suma", "savana", "bara", "jungla", "reka"];

        let tipoviBoja = ["green", "yellow", "red", "orange", "blue"];

        let opcija = null;

        let labela = null;

        let divRb = null;

        tipoviStanista.forEach((staniste, index) => {

            divRb = document.createElement("div");

            opcija = document.createElement("input");

            opcija.type = "radio";

            opcija.name = this.naziv;

            opcija.value = tipoviBoja[index];

            labela = document.createElement("label");

            labela.innerHTML = staniste;

            divRb.appendChild(opcija);

            divRb.appendChild(labela);

            kontForma.appendChild(divRb);

        })

        divRb = document.createElement("div");

        let selX = document.createElement("select");

        labela = document.createElement("label");

        labela.innerHTML = "X:"

        divRb.appendChild(labela);

        divRb.appendChild(selX);

        for (let i = 0; i < this.m; i++) {

            opcija = document.createElement("option");

            opcija.innerHTML = i;

            opcija.value = i;

            selX.appendChild(opcija);

        }

        kontForma.appendChild(divRb);

        let selY = document.createElement("select");

        labela = document.createElement("label");

        labela.innerHTML = "Y:"

        divRb.appendChild(labela);

        divRb.appendChild(selY);

        for (let i = 0; i < this.n; i++) {

            opcija = document.createElement("option");

            opcija.innerHTML = i;

            opcija.value = i;

            selY.appendChild(opcija);

        }

        kontForma.appendChild(divRb);

        const dugme = document.createElement("button");

        dugme.innerHTML = "Dodaj zivotinje";

        kontForma.appendChild(dugme);

        dugme.onclick = (ev) => {

            const vrsta = this.kontejner.querySelector(".vrsta").value;

            const kolicina = parseInt(this.kontejner.querySelector(".kolicina").value);

            const tip = this.kontejner.querySelector(`input[name='${this.naziv}']:checked`);

            if (tip == null)

                alert("Molimo Vas izaberite tip stnaista");

            let x = parseInt(selX.value);

            let y = parseInt(selY.value);

            // Ovde je zamenjena kompletna provera upisa, koja je prebačena na server. Uvek je bolje takve promene vršiti na serveru.

            /\*let potenzijalnaLok = this.lokacije.find(lok => lok.vrsta == vrsta

                && lok.kapacitet + kolicina <= this.kapacitet

                && (lok.x != x || lok.y != y));

            if (potenzijalnaLok)

                alert("Postoji nepopunjena lokacija sa navedenom vestom! Lokacija je (" + potenzijalnaLok.x + "," + potenzijalnaLok.y + ")");

            else

                this.lokacije[x \* this.n + y].azurirajLokaciju(vrsta, kolicina, tip.value, x, y);\*/

            fetch("https://localhost:5001/Zoo/UpisLokacije/" + this.id, {

                method: "POST",

                headers: {

                    "Content-Type": "application/json"

                },

                body: JSON.stringify({

                    vrsta: vrsta,

                    kapacitet: kolicina,

                    maxKapacitet: this.kapacitet,

                    tip: tip.value,

                    x: x,

                    y: y,

                })

            }).then(p => {

                if (p.ok) {

                    this.lokacije[x \* this.n + y].azurirajLokaciju(vrsta, kolicina, tip.value, x, y);

                }

                else if (p.status == 400) {

                    // BadRequest vraća lokaciju kao json. Zato čitamo taj json ispod i upisujemo u greskaLokacija, koju ispisujemo u alert-u.

                    const greskaLokacija = { x: 0, y: 0 };

                    p.json().then(q => {

                        greskaLokacija.x = q.x;

                        greskaLokacija.y = q.y;

                        alert("Postoji nepopunjena lokacija sa navedenom mestom! Lokacija je (" + greskaLokacija.x + "," + greskaLokacija.y + ")");

                    });

                }

                else {

                    alert("Greška prilikom upisa.");

                }

            }).catch(p => {

                alert("Greška prilikom upisa.");

            });

        }

    }

    crtajLokacije(host) {

        const kontLokacije = document.createElement("div");

        kontLokacije.className = "kontLokacije";

        host.appendChild(kontLokacije);

        let red;

        let lok;

        for (let i = 0; i < this.m; i++) {

            red = document.createElement("div");

            red.className = "red";

            kontLokacije.appendChild(red);

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

                //lokacija= document.createElement("div");

                // lokacija.className="lok";

                lok = new Lokacija(i, j, "", "", this.kapacitet);

                this.dodajLokaciju(lok);

                lok.crtajLokaciju(red);

            }

        }

    }

}

export class Lokacija {

    constructor(i, j, tip, vrsta, Maxkapacitet) {

        this.x = i;

        this.y = j;

        this.kapacitet = 0;

        this.maxKapacitet = Maxkapacitet;

        this.tip = tip; //boja stanista

        this.vrsta = vrsta;// lav, labud...

        this.miniKontejner = null;

    }

    vratiBoju() {

        if (!this.tip)

            return "pink";

        else

            return this.tip;

    }

    crtajLokaciju(host) {

        this.miniKontejner = document.createElement("div");

        this.miniKontejner.className = "lok";

        this.miniKontejner.innerHTML = "Prazno, " + this.kapacitet + ", (" + this.maxKapacitet + ")";

        this.miniKontejner.style.backgroundColor = this.vratiBoju();

        host.appendChild(this.miniKontejner);

    }

    azurirajLokaciju(vrsta, kolicina, tip, x, y) {

        console.log("pozvano azuriranje");

        if (kolicina + this.kapacitet > this.maxKapacitet)

            alert("Kapacitet lokacije je popunjen");

        else {

            console.log(kolicina);

            this.vrsta = vrsta;

            this.tip = tip;

            this.kapacitet += kolicina;

            this.miniKontejner.innerHTML = this.vrsta + ", " + this.kapacitet + ", (" + this.maxKapacitet + ")";

            this.miniKontejner.style.backgroundColor = this.vratiBoju();

        }

    }

}

import { Lokacija } from "./lokacija.js";

import { Vrt } from "./vrt.js"

// Pre početka rada aplikacije ne postoji ni jedan zoološki vrt u bazi podataka

// Kao vežbu, kreirati 2 zoološka vrta sa podacima po želji iz JS-a putem fetch metode.

fetch("https://localhost:5001/Zoo/PreuzmiVrtove").then(p => {

    p.json().then(data => {

        data.forEach(vrt => {

            const vrt1 = new Vrt(vrt.id, vrt.naziv, vrt.n, vrt.m, vrt.kapacitet);

            vrt1.crtajVrt(document.body);

            vrt.lokacije.forEach(l => {

                vrt1.lokacije[l.x \* vrt.n + l.y].azurirajLokaciju(l.vrsta, l.kapacitet, l.tip, l.x, l.y);

            });

        });

    });

});

//const vrt1 = new Vrt("Zoo vrt", 5,4,6);

//vrt1.crtajVrt(document.body);

//const vrt2 = new Vrt("Drugi vrt", 10,10,20);

//vrt2.crtajVrt(document.body);

.kontejner{

    display: flex;

    border:  1px solid black;

}

.kontForma{

    display: flex;

    flex-direction: column;

    flex-basis: 200px;

}

.kontLokacije{

    display: flex;

    flex-grow: 1;

    flex-direction: column;

}

.red{

    display: flex;

    flex-grow: 1;

}

.lok{

    border: 1px solid black;

    flex: 1 1 10px;

    margin: 5px;

    text-align: center;

}