const country\_name\_element = document.querySelector(".country .name");

const total\_cases\_element = document.querySelector(".total-cases .value");

const new\_cases\_element = document.querySelector(".total-cases .new-value");

const recovered\_element = document.querySelector(".recovered .value");

const new\_recovered\_element = document.querySelector(".recovered .new-value");

const deaths\_element = document.querySelector(".deaths .value");

const new\_deaths\_element = document.querySelector(".deaths .new-value");

const ctx = document.getElementById("axes\_line\_chart").getContext("2d");

// APP VARIABLES

let app\_data = [],

cases\_list = [],

recovered\_list = [],

deaths\_list = [],

deaths = [],

formatedDates = [];

// GET USERS COUNTRY CODE

let country\_code = geoplugin\_countryCode();

let user\_country;

country\_list.forEach((country) => {

if (country.code == country\_code) {

user\_country = country.name;

}

});

/\* ---------------------------------------------- \*/

/\* FETCH API \*/

/\* ---------------------------------------------- \*/

function fetchData(country) {

user\_country = country;

country\_name\_element.innerHTML = "Loading...";

(cases\_list = []),

(recovered\_list = []),

(deaths\_list = []),

(dates = []),

(formatedDates = []);

var requestOptions = {

method: "GET",

redirect: "follow",

};

const api\_fetch = async (country) => {

await fetch(

"https://api.covid19api.com/total/country/" +

country +

"/status/confirmed",

requestOptions

)

.then((res) => {

return res.json();

})

.then((data) => {

data.forEach((entry) => {

dates.push(entry.Date);

cases\_list.push(entry.Cases);

});

});

await fetch(

"https://api.covid19api.com/total/country/" +

country +

"/status/recovered",

requestOptions

)

.then((res) => {

return res.json();

})

.then((data) => {

data.forEach((entry) => {

recovered\_list.push(entry.Cases);

});

});

await fetch(

"https://api.covid19api.com/total/country/" + country + "/status/deaths",

requestOptions

)

.then((res) => {

return res.json();

})

.then((data) => {

data.forEach((entry) => {

deaths\_list.push(entry.Cases);

});

});

updateUI();

};

api\_fetch(country);

}

fetchData(user\_country);

function updateUI() {

updateStats();

axesLinearChart();

}

function updateStats() {

const total\_cases = cases\_list[cases\_list.length - 1];

const new\_confirmed\_cases = total\_cases - cases\_list[cases\_list.length - 2];

const total\_recovered = recovered\_list[recovered\_list.length - 1];

const new\_recovered\_cases =

total\_recovered - recovered\_list[recovered\_list.length - 2];

const total\_deaths = deaths\_list[deaths\_list.length - 1];

const new\_deaths\_cases = total\_deaths - deaths\_list[deaths\_list.length - 2];

country\_name\_element.innerHTML = user\_country;

total\_cases\_element.innerHTML = total\_cases;

new\_cases\_element.innerHTML = `+${new\_confirmed\_cases}`;

recovered\_element.innerHTML = total\_recovered;

new\_recovered\_element.innerHTML = `+${new\_recovered\_cases}`;

deaths\_element.innerHTML = total\_deaths;

new\_deaths\_element.innerHTML = `+${new\_deaths\_cases}`;

// format dates

dates.forEach((date) => {

formatedDates.push(formatDate(date));

});

}

// UPDATE CHART

let my\_chart;

function axesLinearChart() {

if (my\_chart) {

my\_chart.destroy();

}

my\_chart = new Chart(ctx, {

type: "line",

data: {

datasets: [

{

label: "Cases",

data: cases\_list,

fill: false,

borderColor: "#FFF",

backgroundColor: "#FFF",

borderWidth: 1,

},

{

label: "Recovered",

data: recovered\_list,

fill: false,

borderColor: "#009688",

backgroundColor: "#009688",

borderWidth: 1,

},

{

label: "Deaths",

data: deaths\_list,

fill: false,

borderColor: "#f44336",

backgroundColor: "#f44336",

borderWidth: 1,

},

],

labels: formatedDates,

},

options: {

responsive: true,

maintainAspectRatio: false,

},

});

}

// FORMAT DATES

const monthsNames = [

"Jan",

"Feb",

"Mar",

"Apr",

"May",

"Jun",

"Aug",

"Sep",

"Oct",

"Nov",

"Dec",

];

function formatDate(dateString) {

let date = new Date(dateString);

return `${date.getDate()} ${monthsNames[date.getMonth() - 1]}`;

}