<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Run Chart Application</title>

<link rel="stylesheet" href="styles.css">

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script defer src="script.js"></script>

</head>

<body>

<header>

<h1>Run Chart Application</h1>

<img src="logo.png" alt="Company Logo" class="logo">

</header>

<main>

<section class="date-picker">

<label for="start-date">Start Date:</label>

<input type="date" id="start-date">

<label for="end-date">End Date:</label>

<input type="date" id="end-date">

<button id="load-data">Load Data</button>

</section>

<section class="chart-container">

<canvas id="runChart"></canvas>

</section>

<section class="export-section">

<button id="export-excel">Export to Excel</button>

</section>

</main>

<footer>

<p>&copy; 2024 Manufacturing Inc.</p>

</footer>

</body>

</html>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

display: flex;

flex-direction: column;

min-height: 100vh;

background-color: #f0f0f0;

}

header {

background-color: #2c3e50;

color: white;

padding: 20px;

text-align: center;

position: relative;

}

.logo {

position: absolute;

right: 20px;

top: 20px;

width: 50px;

animation: rotate 10s linear infinite;

}

@keyframes rotate {

0% {

transform: rotate(0deg);

}

100% {

transform: rotate(360deg);

}

}

main {

flex: 1;

padding: 20px;

display: flex;

flex-direction: column;

align-items: center;

}

.date-picker {

margin-bottom: 20px;

}

.chart-container {

width: 80%;

max-width: 800px;

margin: 20px 0;

background: white;

border-radius: 8px;

padding: 20px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

.export-section {

margin-top: 20px;

}

button {

background-color: #3498db;

color: white;

padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s;

}

button:hover {

background-color: #2980b9;

}

footer {

background-color: #2c3e50;

color: white;

text-align: center;

padding: 10px;

}

document.addEventListener('DOMContentLoaded', () => {

const ctx = document.getElementById('runChart').getContext('2d');

let runChart;

document.getElementById('load-data').addEventListener('click', () => {

const startDate = document.getElementById('start-date').value;

const endDate = document.getElementById('end-date').value;

fetch(/api/data?startDate=${startDate}&endDate=${endDate})

.then(response => response.json())

.then(data => {

if (runChart) {

runChart.destroy();

}

runChart = new Chart(ctx, {

type: 'line',

data: {

labels: data.map(d => d.timestamp),

datasets: [{

label: 'Measurements',

data: data.map(d => d.value),

borderColor: 'blue',

borderWidth: 1,

pointBackgroundColor: data.map(d => d.value > d.max || d.value < d.min ? 'red' : 'blue')

}]

},

options: {

scales: {

x: {

type: 'time',

time: {

unit: 'day'

}

}

},

animation: {

duration: 1000,

easing: 'easeInOutBounce'

}

}

});

});

});

document.getElementById('export-excel').addEventListener('click', () => {

fetch('/api/export', { responseType: 'blob' })

.then(response => response.blob())

.then(blob => {

const link = document.createElement('a');

link.href = window.URL.createObjectURL(blob);

link.download = 'data.xlsx';

link.click();

});

});

});

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Run Chart Application</title>

    <link rel="stylesheet" href="styles.css">

    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

    <script defer src="script.js"></script>

</head>

<body>

    <header>

        <h1>Run Chart Application</h1>

        <img src="logo.png" alt="Company Logo" class="logo">

    </header>

    <main>

        <section class="date-picker">

            <label for="start-date">Start Date:</label>

            <input type="date" id="start-date">

            <label for="end-date">End Date:</label>

            <input type="date" id="end-date">

            <button id="load-data">Load Data</button>

        </section>

**Style.css**

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    display: flex;

    flex-direction: column;

    min-height: 100vh;

    background-color: #f0f0f0;

}

header {

    background-color: #2c3e50;

    color: white;

    padding: 20px;

    text-align: center;

    position: relative;

}

.logo {

    position: absolute;

    right: 20px;

    top: 20px;

    width: 50px;

    animation: rotate 10s linear infinite;

}

@keyframes rotate {

    0% {

        transform: rotate(0deg);

    }

    100% {

        transform: rotate(360deg);

    }

}

main {

    flex: 1;

    padding: 20px;

    display: flex;

    flex-direction: column;

    align-items: center;

}

.date-picker {

    margin-bottom: 20px;

}

.chart-container {

    width: 80%;

    max-width: 800px;

    margin: 20px 0;

    background: white;

    border-radius: 8px;

    padding: 20px;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

.export-section {

    margin-top: 20px;

}

button {

    background-color: #3498db;

    color: white;

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    transition: background-color 0.3s;

}

button:hover {

    background-color: #2980b9;

}

footer {

    background-color: #2c3e50;

    color: white;

    text-align: center;

    padding: 10px;

}

**Script.js**

ocument.addEventListener('DOMContentLoaded', () => {

    const ctx = document.getElementById('runChart').getContext('2d');

    let runChart;

    document.getElementById('load-data').addEventListener('click', () => {

        const startDate = document.getElementById('start-date').value;

        const endDate = document.getElementById('end-date').value;

        fetch(/api/data?startDate=${startDate}&endDate=${endDate})

            .then(response => response.json())

            .then(data => {

                if (runChart) {

                    runChart.destroy();

                }

                runChart = new Chart(ctx, {

                    type: 'line',

                    data: {

                        labels: data.map(d => d.timestamp),

                        datasets: [{

                            label: 'Measurements',

                            data: data.map(d => d.value),

                            borderColor: 'blue',

                            borderWidth: 1,

                            pointBackgroundColor: data.map(d => d.value > d.max || d.value < d.min ? 'red' : 'blue')

                        }]

                    },

                    options: {

                        scales: {

                            x: {

                                type: 'time',

                                time: {

                                    unit: 'day'

                                }

                            }

                        },

                        animation: {

                            duration: 1000,

                            easing: 'easeInOutBounce'

                        }

                    }

                });

            });

    });

    document.getElementById('export-excel').addEventListener('click', () => {

        fetch('/api/export', { responseType: 'blob' })

            .then(response => response.blob())

            .then(blob => {

                const link = document.createElement('a');

                link.href = window.URL.createObjectURL(blob);

                link.download = 'data.xlsx';

                link.click();

            });

    });

});