# **CODE:**

# <body>

<div id="container">

<h2>File Size Reducer</h2>

<input type="file" id="fileInput">

<input type="number" id="sizeInput" placeholder="Target size in KB">

<button onclick="compressFile()">Compress & Download</button>

<a id="downloadLink" style="display:none">Download Compressed File</a>

</div>

<script>

async function compressFile() {

const fileInput = document.getElementById("fileInput");

const sizeInput = document.getElementById("sizeInput");

const downloadLink = document.getElementById("downloadLink");

if (!fileInput.files.length || !sizeInput.value) {

alert("Please select a file and target size.");

return;

}

const file = fileInput.files[0];

const targetSize = parseInt(sizeInput.value) \* 1024; // Convert KB to Bytes

// Using compression for images

if (file.type.startsWith("image/")) {

const imageCompression = await import("https://cdn.jsdelivr.net/npm/browser-image-compression/+esm");

const options = {

maxSizeMB: targetSize / (1024 \* 1024),

useWebWorker: true

};

const compressedBlob = await imageCompression.default(file, options);

triggerDownload(compressedBlob, file.name);

}

// Using general file compression for other files

else {

const fflate = await import("https://cdn.jsdelivr.net/npm/fflate/+esm");

fflate.zip({[file.name]: new Uint8Array(await file.arrayBuffer())}, (compressed) => {

triggerDownload(new Blob([compressed]), file.name + ".zip");

});

}

}

function triggerDownload(blob, fileName) {

const downloadLink = document.getElementById("downloadLink");

const url = URL.createObjectURL(blob);

downloadLink.href = url;

downloadLink.download = fileName;

downloadLink.style.display = "block";

downloadLink.click();

}

</script>

</body>