let text = $input.first().json.cleaned\_text || '';

// 1. Remove footer/news block starting from unique footer marker onwards

const footerMarker = '\n\n\_\_ Share';

const footerIndex = text.indexOf(footerMarker);

if (footerIndex !== -1) {

text = text.slice(0, footerIndex);

}

// 2. Split text into lines for filtering

const lines = text.split('\n');

// Define lines to exclude entirely

const excludeLineKeywords = [

'Follow us on',

'Found this article interesting?',

'Subscribe – Get Latest News',

'Home',

'Newsletter',

'Webinars',

'Expert Insights',

'Contact',

'Resources',

'About THN',

'Jobs',

'Advertise with us',

'Contact/Tip Us',

'RSS Feeds',

'Email Alerts',

'javascript:void(0)',

'blob:http://localhost',

'[\_\_]',

'[\_\_]',

'[\_\_]',

'[\_\_]',

'[\_\_]',

'[\_\_]',

'[\_\_]',

'[\_\_]',

'\_\_',

'\_\_',

'\_\_',

'\_\_',

'\_\_',

'\_\_',

'\_\_',

'===============',

'==========================================================================================================================================================================',

'---------------------------------------------------------------',

];

// Domains to block URLs for removal

const blockedDomains = [

'blogger.googleusercontent.com',

'blogger.com',

];

// Helper to remove markdown images

function removeImages(line) {

return line.replace(/\[!\[.\*?\]\(.\*?\)\]\(.\*?\)/g, '').replace(/!\[.\*?\]\(.\*?\)/g, '');

}

// Helper to convert markdown links [text](url) -> text only

function removeLinks(line) {

return line.replace(/\[([^\]]+)\]\((https?:\/\/[^\s)]+)\)/g, '$1');

}

// Remove URLs from blocked domains inside a line

function removeBlockedUrls(line) {

return line.replace(/https?:\/\/[^\s)]+/g, url => {

for (const domain of blockedDomains) {

if (url.includes(domain)) return ''; // remove URL entirely

}

return url; // keep other URLs

}).replace(/\s{2,}/g, ' ').trim();

}

const filteredLines = [];

for (let line of lines) {

line = line.trim();

if (!line) {

// Empty line means paragraph break — keep it

filteredLines.push('');

continue;

}

// Skip lines containing excluded keywords

if (excludeLineKeywords.some(keyword => line.includes(keyword))) {

continue;

}

line = removeImages(line);

line = removeLinks(line);

line = removeBlockedUrls(line);

if (line.length > 0) {

filteredLines.push(line);

}

}

// Join filtered lines with '\n' preserving paragraphs (empty lines),

// then replace all single '\n' (not double) inside paragraphs with spaces

let cleanedContent = filteredLines.join('\n');

cleanedContent = cleanedContent.replace(/([^\n])\n([^\n])/g, '$1 $2').trim();

return [{

json: {

title: $input.first().json.title,

url: $input.first().json.url,

published\_time: $input.first().json.published\_time,

cleaned\_text: cleanedContent,

}

}];