



# Case Study: Tabirly Ekosisteminde "Quick-Cite" Mimarisinin Teknik Analizi

<https://tr.tabirly.com/>

Proje: Tabirly Quick-Cite (QC) — v2025-09-06

Platform: Blogger Altyapısı / Vanilla JS Entegrasyonu

Kapsam: İstemci Taraflı Doğal Dil İşleme (NLP) ve DOM Manipülasyonu

Yazar: Tabirly Teknik Ekibi

## 1. Yönetici Özeti: Problemin Tanımı ve Çözüm

Tabirly gibi spiritüel, felsefi ve derinlikli konuları ele alan platformlarda, bilginin kaynağını doğrulanabilir kılmak hayati önem taşır. Ancak Blogger gibi geleneksel CMS yapılarında, uzun bir makale içindeki yüzlerce atıfı (citation) yönetmek, dipnotları <sup> etiketleriyle manuel olarak eşleştirmek ve bunları kaynakça listesiyle senkronize etmek yazar için hataya açık ve zaman alıcı bir süreçtir.

**Quick-Cite (QC)**, bu süreci otomatize etmek için geliştirilmiş, sadece editör modunda çalışan (?qc=1), tarayıcı tabanlı "akıllı bir asistandır". Bu kod, statik bir HTML sayfasını, üzerinde çalışılabilir dinamik bir veritabanına dönüştürür.

## 2. Genel Mimari ve Konumlandırma

Quick-Cite, Tabirly'nin sunucu tarafına (server-side) yük bindirmeden, tamamen **istemci tarafında (browser context)** çalışacak şekilde tasarlanmıştır.

- **Entegrasyon:** Kod, Blogger'ın şablon yapısına "parazit olmayan" bir katman olarak eklenir. `if(!(q.get('qc')==='1' || preview)) return;` satırı sayesinde, normal ziyaretçiler bu kodun varlığından haberdar olmaz ve site performansı etkilenmez. Sadece editör ?qc=1 parametresiyle girdiğinde "Admin Modu" aktifleşir.
- **Platform Hack (ClickTrap Bypass):** Blogger, taslak önizleme modunda sayfanın üzerine şeffaf bir katman (.blogger-clickTrap) örerek etkileşimi

kısıtlar. Quick-Cite, pointer-events:none CSS enjeksiyonu ile bu katmanı "delerek" editörün çift tıklama eventlerini yakalayabilmesini sağlar. Bu, platformun sınırlarının aşıldığı kritik bir mühendislik detayıdır.

- **Bağımsızlık:** Herhangi bir dış kütüphane (jQuery, React vb.) kullanmaz. Saf (Vanilla) JavaScript ile yazılarak maksimum performans ve sıfır bağımlılık hedeflenmiştir.

```
<details>
```

```
<summary><strong>🔍 İncele: Quick-Cite Ana Motor Kodu (Tam Sürüm)</strong></summary>
```

```
<!-- TB: Quick-Cite (QC) — v2025-09-06 — toggle ?qc=1 -->
```

```
<script id="tb-qc">
```

```
//<![CDATA[
```

```
(function(){
```

```
    var q=new URLSearchParams(location.search), host=location.hostname;
```

```
    var
```

```
    preview=/(^|\.)blogger\.com$/.test(host)||/(^|\.)draft\.blogger\.com$/.test(host)||  
    /blogspot\.\/.test(host);
```

```
    if(!(q.get('qc')==='1'||preview)) return;
```

```
    // Preview clickTrap bypass (tek sefer)
```

```
    if(preview && !document.getElementById('tb-clicktrap-bypass')){
```

```
        var st=document.createElement('style'); st.id='tb-clicktrap-bypass';
```

```
        st.textContent='.blogger-clickTrap,[class*="clickTrap"]{pointer-events:none!important;};';
```

```
        document.head.appendChild(st);
```

```
    }
```

```

// ---- CSS
if(!document.getElementById('tb-cite-css')){
  var css=document.createElement('style'); css.id='tb-cite-css';
  css.textContent = `
:root{ --tb-header-offset:72px; }
#refs li{ scroll-margin-top: var(--tb-header-offset); }
#refs li.tb-ref-highlight{ outline:2px solid rgba(255,193,7,.6);
background:rgba(255,225,130,.35); transition:background .6s; }
#tb-qc{ position:absolute; z-index:10000; background:#111; color:#fff;
font-size:.9rem; border-radius:8px; padding:.35rem .55rem; display:none;
box-shadow:0 10px 30px rgba(0,0,0,.25); }
#tb-qc .row{display:flex;gap:.4rem;align-items:center;margin:.35rem 0}
#tb-qc label{display:flex;gap:.4rem;align-items:flex-start;cursor:pointer}
#tb-qc footer{display:flex;gap:.5rem;justify-content:flex-end;margin-top:.4rem}
.tb-cite-added{animation:tbFlash 1.3s ease}
@keyframes
tbFlash{0%{background:rgba(255,225,130,.5)}100%{background:transparent}}`
;
  document.head.appendChild(css);
}

var isPost = document.body.classList.contains('item-view') ||
document.querySelector('.post-body');
if (!isPost) return;
var body = document.querySelector('.post-body') || document;

```

```

function norm(s){ return (s||'').toString().trim().replace(/\s+/g,' '); }
function deaccent(s){ try{return
(s||'').normalize('NFKD').replace(/[\u0300-\u036f]/g,'');}catch(_){return s;} }
function tok(s){ s=deaccent(norm(s)).toLowerCase().replace(/["'""""""()\s-.,;:!?]/g,'
'); return s.split(/\s+/).filter(Boolean); }
function getText(el){ return norm(el.textContent||''); }

```

```

var refs = body.querySelector('#refs') || (function(){ var
ls=Array.from(body.querySelectorAll('ol,ul')); return ls.length?ls[ls.length-1]:null;
})();
if(!refs) return;

```

```

// anchor üret
Array.from(refs.querySelectorAll('li')).forEach(function(li,i){
var id='kaynak'+(i+1);
if(!document.getElementById(id)){ var s=document.createElement('span');
s.id=id; li.insertBefore(s, li.firstChild); }
});

```

```

function yearOf(t){ var m=t.match(/(?:\D|^)((?:1[4-9]\d{2}|20\d{2}|19\d{2}))/i);
return m?m[1]:''; }

```

```

function titleOf(li){ var e=li.querySelector('em,i'); if(e) return getText(e); var
t=getText(li), p=t.split(' '); return p.length>1?p[1]:t; }

```

```

function authorsOf(t){
t=t.replace(/\bet al\./ig,' ');
var head=t.split('.')[0], frags=head.split(/[,;&]/).map(norm).filter(Boolean),
parts=[];

```

```

var particles=new
Set(['al','bin','b.','van','von','de','da','di','del','della','der','le','la','mac','mc','o','ibn','el','
ve','and','&']);
    frags.forEach(function(p){ var ws=p.split(/\s+/); if(!ws.length) return; var
i=ws.length-1, pick=[ws[i]]; i--;
while(i>=0&&particles.has(ws[i].toLowerCase())){pick.unshift(ws[i]); i--;}
parts.push(pick.join(' ')); });
    if(!parts.length){ var
m=t.match(/^\s*([A-ZÇĞİÖŞÜ][A-Za-zÇĞİÖŞÜçğışöşü'\-]+(?:\s+[A-ZÇĞİÖŞÜa-z
çğışöşü'\-]+){0,3})/); if(m) parts=[m[1]]; }
    var seen={}, out=[]; parts.forEach(function(a){ a=norm(a);
if(a&&!seen[a]){seen[a]=1; out.push(a);} }); return out.slice(0,3);
}

```

```

var items = Array.from(refs.querySelectorAll('li'));
var refIndex = items.map(function(li, idx){
    var full=getText(li), title=titleOf(li), authors=authorsOf(full);
    return { n:idx+1, anchor:'kaynak'+(idx+1), title, authors, year:yearOf(full),
titleToks:tok(title), authorToks:tok(authors.join(' ')) };
});

```

```

function jaccard(a,b){ var A=new Set(a),B=new Set(b),i=0;
A.forEach(x=>{if(B.has(x)) i++;}); var u=A.size+B.size-i; return u? i/u : 0; }
function extendYearRight(r){ var end=r.endContainer;
if(end&&end.nodeType===3){ var tail=end.nodeValue.slice(r.endOffset,
r.endOffset+8), m=tail.match(/^\s*(1[4-9]\d{2}|20\d{2}|19\d{2})/); return
m?m[1]:'';} return ''; }

```

```

function rankCandidates(query){
    var q=norm(query), m=q.match(/(?:^|\D)((?:1[4-9]\d{2}|20\d{2}|19\d{2}))/),
    year=m?m[1]:'', qPure=year?q.replace(m[0],' '):q, qt=tok(qPure);
    var isAuthorish = qt.length<=2, wT=isAuthorish?0.2:0.6,
    wA=isAuthorish?0.7:0.3, wY=1-(wT+wA);
    var scores=[]; refIndex.forEach(function(r){
        var ts=jaccard(qt, r.titleToks), as=jaccard(qt, r.authorToks),
        ys=(year&&year)?(year===r.year?1:(year.replace(/\D/g,'')==r.year.replace(/\D/g,
        '?.6:0)):0;
        scores.push([r.n, wT*ts + wA*as + wY*ys, r]);
    });
    scores.sort((a,b)=>b[1]-a[1]); return scores.slice(0,10);
}

```

```

// UI balonu
var bubble=(function(){ var d=document.createElement('div'); d.id='tb-qc';
    d.innerHTML='<div class="body"></div><footer><button type="button"
id="tb-qc-cancel">Kapat</button><button type="button"
id="tb-qc-apply">Ekle</button></footer>';
    document.body.appendChild(d);
    d.querySelector('#tb-qc-cancel').onclick=(()=>d.style.display='none'; return d;
})();

```

```

function makeSup(n){ var a=document.createElement('a'); a.href='#kaynak'+n;
a.textContent=String(n); var sup=document.createElement('sup');
sup.className='tb-cite-added'; sup.appendChild(a); return sup; }

```

```

function insertOnce(range, n){ try{ var sup=makeSup(n),
r=range.cloneRange(); r.collapse(false); r.insertNode(sup);
sup.scrollIntoView({block:'nearest'});}catch(e){} }

function insertForAll(needle, n){
  var nd=norm(needle).toLowerCase();
  var walker=document.createTreeWalker(body, NodeFilter.SHOW_TEXT,
{acceptNode:function(node){
  if(!node.nodeValue) return NodeFilter.FILTER_REJECT;
  if(node.parentElement.closest('#refs')) return NodeFilter.FILTER_REJECT;
  return node.nodeValue.toLowerCase().includes(nd) ?
NodeFilter.FILTER_ACCEPT : NodeFilter.FILTER_REJECT;
}});
  var nodes=[]; while(walker.nextNode()) nodes.push(walker.currentNode);
  nodes.forEach(function(node){
    var txt=node.nodeValue, idx=txt.toLowerCase().indexOf(nd); if(idx<0) return;
    var before=document.createTextNode(txt.slice(0, idx+nd.length)),
after=document.createTextNode(txt.slice(idx+nd.length));
    var sup=makeSup(n), frag=document.createDocumentFragment();
    frag.appendChild(before); frag.appendChild(sup); frag.appendChild(after);
    node.parentNode.replaceChild(frag, node);
  });
}

```

```

function showBubble(rect, query, ranked, range){
  var bodyEl=bubble.querySelector('.body'); bodyEl.innerHTML="";
  bodyEl.insertAdjacentHTML('beforeend','<div><strong>"+query+"</strong>
için eşleşme seç:</div>');

```



```

if (!ranked.length){
    bodyEl.insertAdjacentHTML('beforeend','<div>Aday bulunamadı. Kaynakçayı
kontrol et.</div>');
    bubble.querySelector('#tb-qc-apply').onclick={()=>{
bubble.style.display='none'; };
    } else {
        ranked.forEach(function([n,score,r],i){
            bodyEl.insertAdjacentHTML('beforeend',
                '<div class="row"><label><input type="radio" name="tb-qc-choice"
value="" +n+"" '+ (i===0?'checked':'') + '>'+
                '<span><strong>'+(r.title||'(başlık yok)')+</strong><br><span
style="opacity:.75">'+(r.authors.join(', ')||'')+' — '+ (r.year||'yıl yok')+ ' • skor
'+score.toFixed(2)+'</span></span></label></div>');
            });
            bodyEl.insertAdjacentHTML('beforeend','<div class="row"><label><input
type="checkbox" id="tb-qc-all"> Bu ifadeye tüm metinde ekle</label></div>');
            bubble.querySelector('#tb-qc-apply').onclick=function(){
                var chosen=bubble.querySelector('input[name="tb-qc-choice"]:checked');
                if(!chosen){ bubble.style.display='none'; return; }
                var n=parseInt(chosen.value,10),
                applyAll=document.getElementById('tb-qc-all').checked;
                if (applyAll) insertForAll(query, n); else insertOnce(range, n);
                bubble.style.display='none';
            };
        }

        var x=Math.min(window.innerWidth-10, Math.max(10, rect.left +
window.scrollX)), y=rect.bottom + window.scrollY + 8;

```

```
    bubble.style.left=x+'px'; bubble.style.top=y+'px'; bubble.style.display='block';  
}
```

```
body.addEventListener('dblclick', function(){  
    var sel=window.getSelection(); if(!sel||sel.rangeCount===0) return;  
    var r=sel.getRangeAt(0).cloneRange(); var s=norm(sel.toString());  
    if(!s||s.length>120) return;  
    var yr=extendYearRight(r); if(yr) s=s+' '+yr;  
    var ranked=rankCandidates(s); var rect=r.getBoundingClientRect();  
    showBubble(rect, s, ranked, r);  
});
```

```
// Undo  
(function(){  
    var stack=[], mo=new MutationObserver(function(muts){  
        muts.forEach(function(m){ m.addedNodes &&  
m.addedNodes.forEach(function(n){ if(n.nodeType===1 && n.classList &&  
n.classList.contains('tb-cite-added')) stack.push(n); }); });  
    });  
    mo.observe(document.body,{childList:true,subtree:true});  
    document.addEventListener('keydown', function(e){  
        if((e.ctrlKey||e.metaKey)&&!e.shiftKey&&e.key.toLowerCase()=== 'z'){ var  
el=stack.pop(); if(el&&el.parentNode) el.parentNode.removeChild(el); }  
    });  
})();
```

```
// Memory (apply-all kalıcı)
```

```

(function(){
    var STORE='tb_qc_map::'+(location.pathname||location.href);
    function nrm(s){ return (s||'').toString().trim().toLowerCase().replace(/\s+/g,' ');
}

    function load(){ try{return
JSON.parse(localStorage.getItem(STORE)||'{}');}catch(_){return {} }

    function save(m){ try{localStorage.setItem(STORE,
JSON.stringify(m));}catch(_){ } }

    function insertAll(needle, n){
        var bodyEl=document.querySelector('.post-body')||document;
        needle=nrm(needle);
        var walker=document.createTreeWalker(bodyEl, NodeFilter.SHOW_TEXT,
{acceptNode:function(node){
            if(!node.nodeValue) return NodeFilter.FILTER_REJECT;
            if(node.parentElement.closest('#refs')) return NodeFilter.FILTER_REJECT;
            return node.nodeValue.toLowerCase().includes(needle) ?
NodeFilter.FILTER_ACCEPT : NodeFilter.FILTER_REJECT;
        }});
        var nodes=[]; while(walker.nextNode()) nodes.push(walker.currentNode);
        nodes.forEach(function(node){
            var txt=node.nodeValue, low=txt.toLowerCase(), idx=low.indexOf(needle);
            if(idx<0) return;
            if (idx+needle.length===txt.length){ var ns=node.nextSibling; if(ns &&
ns.nodeType===1 && ns.tagName==='SUP' &&
ns.querySelector('a[href="#kaynak'+n+'"']')) return; }
            var before=document.createTextNode(txt.slice(0, idx+needle.length)),
            after=document.createTextNode(txt.slice(idx+needle.length));

```

```

        var a=document.createElement('a'); a.href='#kaynak'+n;
a.textContent=String(n); var sup=document.createElement('sup');
sup.appendChild(a);
        var frag=document.createDocumentFragment(); frag.appendChild(before);
frag.appendChild(sup); frag.appendChild(after);
        node.parentNode.replaceChild(frag, node);
    });
}
var map=load(); Object.keys(map).forEach(function(k){ insertAll(k, map[k]);
});
document.addEventListener('click', function(e){
    if(!e.target.closest('#tb-qc-apply')) return;
    var bubble=document.getElementById('tb-qc'); if(!bubble) return;
    var strong=bubble.querySelector('.body strong'); var
raw=strong?strong.textContent.replace(/["']/g,'"');
    var chosen=bubble.querySelector('input[name="tb-qc-choice"]:checked');
var n=chosen?parseInt(chosen.value,10):null;
    var applyAllEl=bubble.querySelector('#tb-qc-all'); var
applyAll=applyAllEl?applyAllEl.checked:false;
    if(!raw||!n) return;
    if(applyAll){ var m=load(); m[nrm(raw)]=n; save(m); }
    }, false);
})();

// Export butonu
(function(){
    var btn=document.createElement('button');

```

```
btn.id='tb-export'; btn.textContent='HTML'e Kaydet';

btn.style.cssText='position:fixed;right:16px;bottom:16px;z-index:10001;padding:
.55rem
.75rem;border:0;border-radius:10px;background:#111;color:#fff;box-shadow:0
10px 30px rgba(0,0,0,.25);cursor:pointer';
    document.body.appendChild(btn);
    btn.addEventListener('click', function(){
        var root=document.createElement('div');
        var
clone=(document.querySelector('.post-body')||document.body).cloneNode(true);
        clone.querySelectorAll('#tb-qc, .tb-ref-highlight').forEach(el=>el.remove());

clone.querySelectorAll('sup.tb-cite-added').forEach(el=>el.classList.remove('tb
-cite-added'));
        var
refs=clone.querySelector('#refs')||Array.from(clone.querySelectorAll('ol,ul')).slice(-1)[0];
        if(!refs){ alert('Kaynakça listesi bulunamadı. '); return; }
        root.appendChild(clone);
        var
html=root.innerHTML.replace(/sdata-[^\s]+="[^"]+"/g,"").replace(/sclass=""/g,
');
        navigator.clipboard.writeText(html).catch(()=>{});
        var blob=new Blob([html], {type:'text/html;charset=utf-8'}),
a=document.createElement('a');
```

```
    a.href=URL.createObjectURL(blob); a.download='post-body-updated.html';
document.body.appendChild(a); a.click(); a.remove();
    alert('Panoya kopyalandı ve .html indirildi.');
```

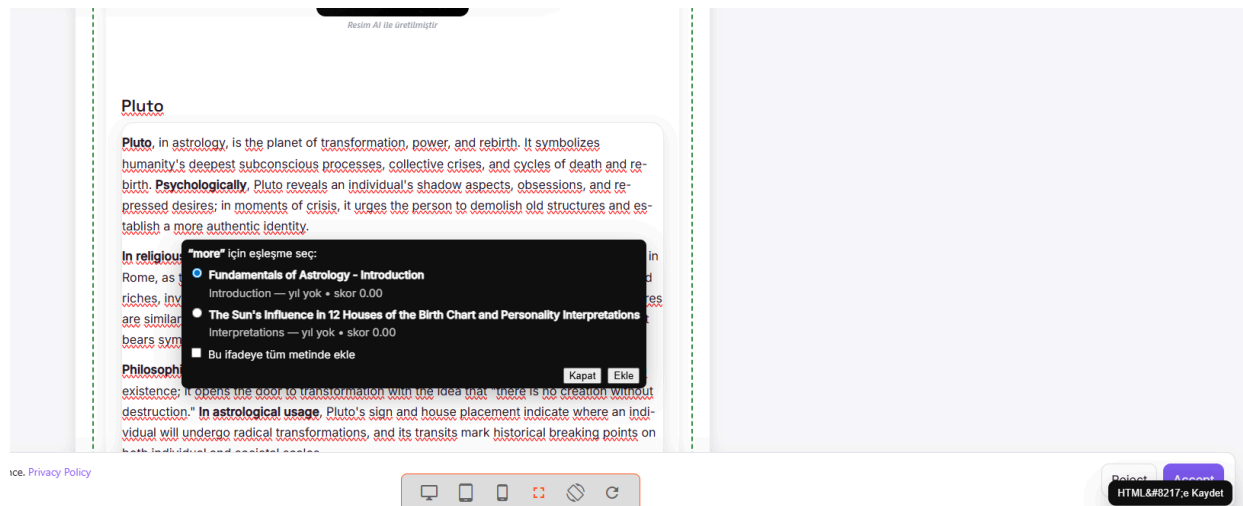
```
});
})();

// SmoothScroll
(function(){
    var bodyEl=document.querySelector('.post-body') || document;
    var refs=bodyEl.querySelector('#refs')||(function(){ var
ls=Array.from(bodyEl.querySelectorAll('ol,ul')); return
ls.length?ls[ls.length-1]:null; })();
    if(!refs) return;
    Array.from(refs.querySelectorAll('li')).forEach(function(li,i){
        var id='kaynak'+(i+1); if(!document.getElementById(id)){ var
s=document.createElement('span'); s.id=id; li.insertBefore(s, li.firstChild); }
    });
    function topFixedOffset(){
        var els=Array.from(document.body.querySelectorAll('*')).slice(0,500);
        for(var el of els){ var cs=getComputedStyle(el); if(cs.position==='fixed' &&
parseInt(cs.top||'0',10)===0 && el.offsetHeight>40) return el.offsetHeight+8; }
        var
v=getComputedStyle(document.documentElement).getPropertyValue('--tb-header-offset').trim(); return v?parseInt(v,10):0;
    }
    function smoothFocus(target){
        var y=target.getBoundingClientRect().top + window.scrollY -
```

```

topFixedOffset();
    window.scrollTo({ top:y, behavior:'smooth' });
    var li=target.closest('li'); if(li){ li.classList.add('tb-ref-highlight');
setTimeout( ()=>li.classList.remove('tb-ref-highlight'),1600); }
    }
    document.addEventListener('click', function(e){
        var a=e.target.closest('a[href^="#kaynak"]'); if(!a) return;
        var id=a.getAttribute('href').slice(1), tgt=document.getElementById(id);
        if(!tgt) return;
        e.preventDefault(); smoothFocus(tgt); try{ history.replaceState(null,"'#'+id);
        }catch(_){}
        }, {passive:false});
    })();
})();
//]]>
</script>
</details>

```



### 3. İş Akışı: Veriden Çıktıya Adım Adım Analiz

Sistemin çalışma prensibi, bir veritabanı sorgu motorunun çalışma mantığına benzer. Aşağıdaki şema sistemin temel döngüsünü özetlemektedir:

graph TD

```
A[Sayfa Yüklmesi] -->|?qc=1| B[HTML Parsing & İndeksleme]
B --> C{Kullanıcı Eylemi}
C -->|Çift Tıklama| D[Metin Seçimi & Normalizasyon]
D --> E[Fuzzy Search & Puanlama]
E --> F[Arayüz: Bubble UI Gösterimi]
F --> G{Yazar Seçimi}
G -->|Tek Ekle| H[Range API ile SUP Ekleme]
G -->|Tümüne Ekle| I[TreeWalker ile Toplu İşlem]
H --> J[DOM Güncellemesi]
I --> J
J --> L[UX Katmanı: Polishing]
L --> K[Export: Temiz HTML Çıktısı]
```

#### A. Veri Hazırlığı ve İndeksleme (Indexing)

Sayfa yüklendiğinde script ilk olarak **HTML Parsing** işlemini başlatır.

1. **Hedef Tespiti:** #refs ID'sine sahip veya metin sonundaki ol/ul listelerini (Kaynakça) bulur.
2. **Gelişmiş Yazar Ayırıştırma (Particle Detection):** Sadece isimleri değil, soyisimlerdeki "soyluluk" eklerini de tanır. authorsOf fonksiyonu; "bin",



"van", "de", "von", "mc" gibi parçacıkları algılayarak "De Wart" gibi isimleri tek bir bütün (entity) olarak işler.

3. **Metadata Çıkarımı:** Regex (Düzenli İfadeler) kullanılarak listedeki her maddeden Yazar, Yıl ve Başlık bilgisi ayrıştırılarak refIndex belleğine atılır.

## B. Kullanıcı Etkileşimi ve Sorgu (Querying)

Yazar, metin üzerinde bir ifadeye **çift tıkladığında** (dblclick event listener) süreç tetiklenir:

1. **Seçimi Algılama:** Seçilen metin alınır. Eğer kullanıcı bir cümlelin sonunu seçtiyse ve orada bir yıl varsa (extendYearRight), algoritma akıllıca davranıp yılı da sorguya dahil eder.
2. **Fuzzy Matching (Bulanık Mantık):** Burası sistemin beynidir (rankCandidates). Seçilen metin, kaynakçadaki verilerle **Jaccard Benzerlik İndeksi** kullanılarak kıyaslanır.
  - o *Ağırlıklı Skorlama:* Algoritma körlemesine eşleştirme yapmaz. Eğer sorgu kısaysa (yazar adı gibi), yazar eşleşmesine **0.7**, başlığa **0.2** puan verir.

## C. Arayüz ve Manipölasyon (The Bubble UI)

Kullanıcının seçtiği metnin hemen üzerinde dinamik bir **"Action Bubble"** belirir. Yazar doğru kaynağı seçip "Ekle" dediğinde, Range API kullanılarak seçilen metin bozulmadan yanına `<sup><a href="#kaynak1">1</a></sup>` etiketi "cerrahi" bir hassasiyetle enjekte edilir.

## D. Otomasyon ve Bellek (Memory & Batch Processing)

Kodun en güçlü yanlarından biri **Toplu İşlem** yeteneğidir.

- **"Tüm metinde ekle":** Yazar bu kutucuğu işaretlerse, TreeWalker API tüm makaleyi tarar. Aynı yazar isminin veya kitap adının geçtiği *her yeri* bulur ve

otomatik olarak atıf ekler.

- **Güvenlik Duvarı (Duplicate Guard):** Sistem, otomatik ekleme yaparken kelimenin hemen yanında zaten bir atıf olup olmadığını kontrol eder (node.nextSibling). Bu, Gurdjieff[1][1] gibi tekrarlı ve hatalı atıfların oluşmasını engeller.
- **LocalStorage Kalıcılığı:** Memory modülü, kullanıcının yaptığı "toplu ekleme" tercihlerini tarayıcı hafızasına (localStorage) kaydeder.

## E. Temizleme ve Dışa Aktarım (Export)

İş bittiğinde, "HTML'e Kaydet" butonu devreye girer. Sistem, eklediği CSS'leri, JS listener'larını ve arayüz elemanlarını temizler (DOM Cleaning). Sadece saf, temiz HTML kodunu panoya kopyalar ve .html dosyası olarak indirir.

## 4. Kullanıcı Deneyimi: Adım Adım Senaryo (Walkthrough)

Teknik altyapının pratik hayatta nasıl karşılık bulduğunu göstermek için, bir editörün "Gurdjieff ve 4. Yol" başlıklı bir makaleyi düzenlediği senaryoyu izleyelim:

1. **Aktivasyon:** Yazar, Blogger taslak önizleme sayfasının adres çubuğuna &qc=1 ekler ve Enter'a basar. Bu "Gizli Anahtar", Quick-Cite modülünü uyandırır.
2. **Etkileşim:** Metni okurken "İnsanın Olası Evrimi Psikolojisi" ibaresini görür ve üzerine çift tıklar.
3. **Anlık Analiz:** Sistem milisaniyeler içinde sayfanın altındaki kaynakçayı tarar. "Psikoloji", "Evrin" ve "Olası" kelimelerini içeren kitapları puanlar.
4. **Karar Anı (The Bubble):** İmlecin tepesinde siyah bir balon belirir:
  - *Seçenek A:* P.D. Ouspensky - İnsanın Olası Evrimi Psikolojisi (%95

Eşleşme)

- Seçenek B: Darwin - Türlerin Kökeni (%10 Eşleşme)

- Otomasyon:** Yazar, Seçenek A'yı işaretler. Eğer bu kitap metinde çokça geçiyorsa, "**Bu ifadeye tüm metinde ekle**" kutucuğunu da seçer ve "Ekle" butonuna basar.
- Sonuç:** Sayfadaki tüm ilgili ifadelerin yanına nazikçe bir [4] (veya ilgili sıra numarası) eklenir. UX katmanı sayesinde bu numara kelimeye yapışmaz, zarif bir boşlukla ayrılır.
- Çıktı:** Düzenleme bitince sağ alttaki "HTML'e Kaydet" butonuyla temiz kod indirilir ve Blogger paneline yapıştırılır.

## 5. Kullanılan Teknolojiler ve Tercih Nedenleri

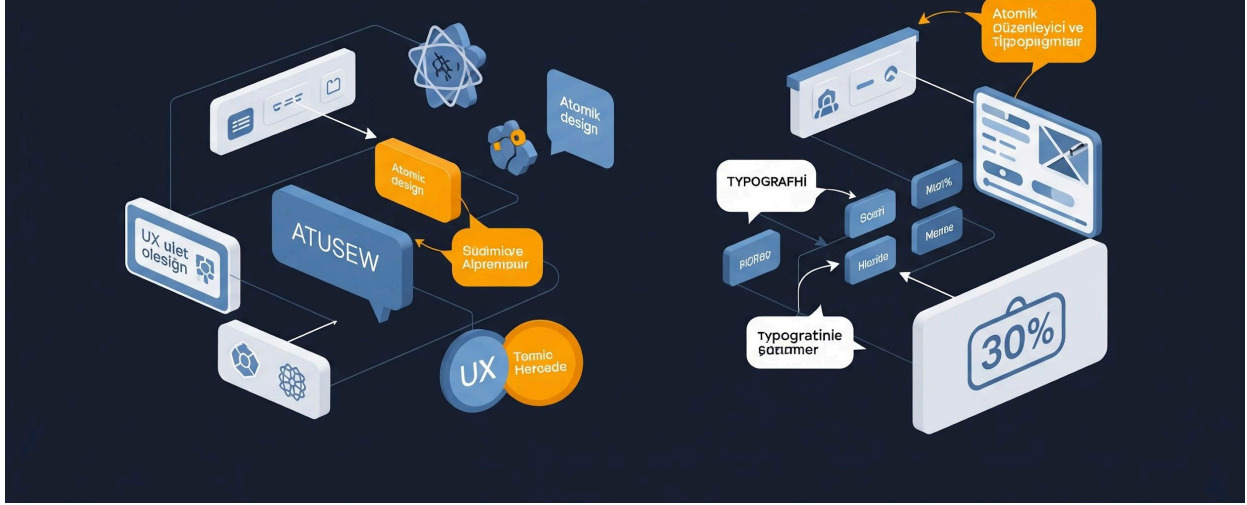
Teknoloji	Kullanım Alanı	Neden Tercih Edildi?
Vanilla JS (ES6+)	Tüm mantık	Framework yükü (React/Vue) olmadan anlık açılış hızı ve DOM üzerinde tam kontrol.
TreeWalker API	DOM Tarama	innerHTML ile metin değiştirmek HTML yapısını bozar (event listenerları siler). TreeWalker sadece "metin düğümlerini" güvenli değiştirir.
Jaccard Similarity	Arama Algoritması	Levenshtein gibi pahalı

		algoritmalar yerine, küme kesişimini baz alan Jaccard, tarayıcıda çok daha hızlı çalışır ve kelime sırasından bağımsızdır.
<b>Range API</b>	Metin İşleme	Kullanıcının seçtiği metnin tam koordinatlarını bulmak ve HTML yapısını bozmadan araya element sıkıştırmak için.
<b>MutationObserver</b>	Dinamik İzleme	Sonradan eklenen (injected) içerikleri yakalayıp stil ve davranış kurallarını anında uygulamak için.

## 6. UX Katmanı: Atomik Düzenleyici ve Tipografi (The Polisher)

## UX KATMANI: ATOMİK DÜZENLEYİCİ VE TİPOGRAFI

(The Polisher)



Sistemin mükemmelleşmesini sağlayan son katman, "Refinement Gadget" olarak adlandırılan tamamlayıcı scripttir. Quick-Cite atıfları eklerken, bu katman editör deneyimini optimize eder:

- 1. Tipografik İyileştirme (Hair Space):** Atıf numaralarının kelimeye yapışmasını engellemek için kod, otomatik olarak görünmez bir "ince boşluk" (\u200A) enjekte eder.
  - Öncesi: ...farkındalık[1] (Sıkışık)
  - Sonrası: ...farkındalık[1] (Nefes alan yapı)
- 2. Atomik Veri Koruma:** Atıflar contenteditable="false" özelliği ile kilitlenir. Bu sayede yazar yanlışlıkla numaranın içine ([1abc]) yazı yazamaz.
- 3. Akıllı Silme (Backspace Logic):** İmleç atıfın yanındayken silme tuşuna basıldığında, kod devreye girer ve atıfı harf harf değil, tek bir "obje" gibi blok halinde siler. Bu, HTML yapısının bozulmasını engeller.

<details>

<summary><strong>🎨 İncele: UX & Tipografi Cilasası Kodu</strong></summary>

<style>

/\* Superscript'ler kelimeye yapışmasın \*/

.tb-cite { margin-left: .15em; }

</style>

<script type="text/javascript">

//<![CDATA[

(function(){

// 1) Tüm .tb-cite'ları normalize et: atomik + öncesine ince boşluk (hair space)

function normalizeCites(root){

(root || document).querySelectorAll('.tb-cite').forEach(function(sup){

// Atomik yap: yanlışlıkla içine yazı eklenmesin

sup.setAttribute('contenteditable','false');

// Öncesi boşlukla ayrılmış mı? Değilse ince boşluk ekle

var prev = sup.previousSibling;

var needsSpace = true;

if(prev && prev.nodeType === 3){

needsSpace = !(/\\s\$/).test(prev.nodeValue));

}

if(needsSpace){

sup.parentNode.insertBefore(document.createTextNode('\\u200A'), sup); //

hair space

}

});

}

// 2) Backspace/Delete davranışı: imleç cite sınırındaysa sadece cite silinsin

```
function keyFix(e){  
  var sel = window.getSelection();  
  if(!sel || !sel.isCollapsed) return; // sadece tek nokta imleçte  
  var node = sel.anchorNode, offset = sel.anchorOffset;
```

// Text node içinde misin?

```
if(!node || node.nodeType !== 3) return;
```

```
var parent = node.parentNode;
```

```
var before = null, after = null;
```

// Offset konumuna göre komşu düğümleri bul

```
if(e.key === 'Backspace'){  
  // imleç satırın başında ise önceki kardeşe bak  
  if(offset === 0){  
    before = parent.previousSibling;  
  }else{  
    // metnin içinde: imleçten önce ayrı bir sup yoksa bırak  
    return;  
  }  
} else if(e.key === 'Delete'){  
  // imleç sonda ise sonraki kardeşe bak  
  if(offset === node.nodeValue.length){  
    after = parent.nextSibling;  
  }else{
```

```
    return;  
  }  
  } else {  
    return;  
  }
```

```
var targetSup = null;  
if(before && before.nodeType === 1 && before.matches('.tb-cite')) targetSup  
= before;  
if(after && after.nodeType === 1 && after.matches('.tb-cite')) targetSup =  
after;
```

```
if(targetSup){  
  e.preventDefault();  
  // Önündeki olası hair space'i de temizle  
  var ps = targetSup.previousSibling;  
  if(ps && ps.nodeType===3 && /\u200A$/.test(ps.nodeValue)){  
    ps.nodeValue = ps.nodeValue.replace(/\u200A$/, '');  
  }  
  targetSup.remove();  
}  
}
```

```
// İlk yüklemede ve sonradan eklenenlerde uygula  
function run(){ normalizeCites(document); }  
if(document.readyState==='loading'){  
document.addEventListener('DOMContentLoaded', run); } else { run(); }
```



```
document.addEventListener('keydown', keyFix, true);

// Dinamik içerik yüklenirse de normalize et
new MutationObserver(function(muts){
  var touched=false;
  muts.forEach(function(m){ if(m.addedNodes && m.addedNodes.length)
touched=true; });
  if(touched) normalizeCites(document);
}).observe(document.documentElement, {childList:true, subtree:true});
})();
//]]>
</script>
</details>
```

## 7. Somut Faydalar: Neden Önemli?

1. **Hata Payının Sıfırlanması:** Manuel yazımda "kaynak 4"e atıf yaparken yanlışlıkla "kaynak 5" yazma riski vardır. Quick-Cite, ID bazlı çalıştığı (#kaynakN) için linkler her zaman doğru hedefe gider.
2. **%80 Zaman Tasarrufu:** 3000 kelimelik, 50 atıflı bir makalede her atıfı tek tek elle kodlamak saatler sürer. "Tümüne Ekle" özelliği ile bu süre dakikalara iner.
3. **Akışın Korunması:** Yazar, HTML kodları arasında boğulmaz. Görsel arayüzde (WYSIWYG) kalır, sadece çift tıklar ve seçer. Düşünce akışı bölünmez.
4. **Otomatik Linkleme (Smooth Scroll):** Kod, aynı zamanda okuyucu deneyimi için scroll-margin-top ve CSS highlight efektlerini de otomatik ekler. Okuyucu bir atıfa tıkladığında, sayfa yumuşakça aşağı kayar ve ilgili

kaynak sarı renkle parlar.

## 8. Sonuç

Tabirly için yazılan bu kod parçası, teknik bir scriptten ziyade, **içerik üretim sürecine entegre edilmiş bir mikro-uygulamadır.**

Şu anki haliyle Quick-Cite; karmaşık, akademik formatlı içerikleri Blogger gibi basit bir altyapıda bile profesyonelce yönetmeyi mümkün kılan, **ölçeklenebilir ve sürdürülebilir** bir mühendislik çözümüdür.

<https://tr.tabirly.com/>