**How to plug FlickWord in via a link (no index surgery)**

**1) Put the game somewhere separate**

* Create a folder: /features/
* Save the full game file from the canvas as:  
  /features/flickword.html
* That’s it. No bundling. No dependency drama.

Kill switch: if FlickWord ever misbehaves, delete /features/flickword.html or hide the link (flag below) and your app stays alive.

**2) Add a simple “Daily Competition” link to your header**

Drop this near your other header buttons (where the old Mardi button used to be). It opens in a new tab.

<!-- In header actions -->

<a id="dailyCompLink"

class="btn secondary"

href="/features/flickword.html"

target="\_blank"

rel="noopener"

title="Daily Competition">

FlickWord

</a>

Styling matches your current buttons; change class if you want the “white/black” look instead.

**3) Optional: append today’s seed in the URL**

If you want the daily word to be locked to a specific calendar day (so cache/timezone nonsense doesn’t change it mid-day), append a ?date= param. Put this anywhere in your existing <script> (after DOM is ready):

<script>

(function wireFlickWordLink(){

const link = document.getElementById('dailyCompLink');

if (!link) return;

// Build a YYYY-MM-DD in UTC to keep all users on the same “daily” word

const now = new Date();

const yyyy = now.getUTCFullYear();

const mm = String(now.getUTCMonth() + 1).padStart(2, '0');

const dd = String(now.getUTCDate()).padStart(2, '0');

const iso = `${yyyy}-${mm}-${dd}`;

const base = '/features/flickword.html';

const url = `${base}?date=${iso}`;

link.href = url;

})();

</script>

Inside flickword.html, you can read ?date= if you want (optional). Your current daily logic based on days-since-epoch will still work fine without it.

**4) Add a visible countdown next to the link (nice touch)**

This gives the “daily” vibe without integrating the game. Drop this near the link:

<span id="dailyCountdown"

style="font-weight:700; padding:6px 10px; border:2px solid var(--border); border-radius:10px; background:var(--card);">

00:00:00

</span>

<script>

(function startDailyCountdown(){

const el = document.getElementById('dailyCountdown');

if (!el) return;

function tick(){

const now = new Date();

const reset = new Date(now);

// Reset at 00:00 UTC so everyone shares the same daily word window

reset.setUTCHours(24,0,0,0);

const diff = Math.max(0, Math.floor((reset - now) / 1000));

const h = String(Math.floor(diff / 3600)).padStart(2, '0');

const m = String(Math.floor((diff % 3600) / 60)).padStart(2, '0');

const s = String(diff % 60).padStart(2, '0');

el.textContent = `⏱ ${h}:${m}:${s}`;

}

tick();

setInterval(tick, 1000);

})();

</script>

Change to local midnight if you prefer America/New\_York, but UTC is simpler for a global audience.

**5) Feature flag (so you can yank it instantly)**

Somewhere in your existing settings boot code (you already simulate Pro/etc.), gate the link:

<script>

(function flickWordFlag(){

// Example flag; set to false to hide

const enableFlickWord = true; // flip to false to “remove” the feature instantly

const el = document.getElementById('dailyCompLink');

const countdown = document.getElementById('dailyCountdown');

if (!el) return;

if (!enableFlickWord) {

el.style.display = 'none';

if (countdown) countdown.style.display = 'none';

}

})();

</script>

Tie it to your existing Pro toggle if you want to make the **link visible to all** but **leaderboards Pro-only** later.

**6) (Optional) Open in a lightweight modal instead of a new tab**

If you want to keep users inside your app, but still isolate the game, use an **iframe modal** that points to the external page. Add this once to your index:

<!-- Modal shell (hidden by default) -->

<div id="fwModal" style="display:none; position:fixed; inset:0; background:rgba(0,0,0,.6); z-index:10000; align-items:center; justify-content:center;">

<div style="background:#fff; width:min(96vw, 560px); height:min(92vh, 760px); border-radius:14px; border:2px solid #000; position:relative; overflow:hidden;">

<button id="fwClose" aria-label="Close"

style="position:absolute; top:8px; right:8px; border:2px solid #000; background:#fff; padding:6px 10px; border-radius:8px; cursor:pointer;">

Exit

</button>

<iframe id="fwFrame" src="" title="FlickWord"

style="width:100%; height:100%; border:0;"></iframe>

</div>

</div>

<script>

(function modalBridge(){

const link = document.getElementById('dailyCompLink');

const modal = document.getElementById('fwModal');

const frame = document.getElementById('fwFrame');

const close = document.getElementById('fwClose');

if (!link || !modal || !frame || !close) return;

// If you want modal behavior instead of new tab, uncomment the listener below

// link.removeAttribute('target'); // optional

// link.addEventListener('click', (e) => {

// e.preventDefault();

// frame.src = link.href;

// modal.style.display = 'flex';

// });

close.addEventListener('click', () => {

modal.style.display = 'none';

frame.src = ''; // unload the game

});

})();

</script>

Toggle the comment to switch between **new tab** and **modal** without touching the game file.

**7) (Future-proofing) Add a tiny bridge for scores**

If/when you want FlickWord to send results back to your app, use postMessage from /features/flickword.html and listen for it in your index:

**In /features/flickword.html (after a win/loss):**

// window.parent may be your app if opened in iframe; in a new tab, this is ignored

try {

window.parent.postMessage(

{ type: 'flickword:result', date: new Date().toISOString().slice(0,10), won: true, guesses: game.guesses.length },

'\*'

);

} catch(\_) {}

**In your index page:**

<script>

window.addEventListener('message', (e) => {

const msg = e.data || {};

if (msg.type === 'flickword:result') {

// TODO: save to Firestore/localStorage/whatever

// e.g., localStorage.setItem('flickword:last', JSON.stringify(msg));

// and maybe flash a toast

console.log('FlickWord result:', msg);

}

});

</script>

No coupling; if you never use it, nothing breaks.

**8) Rollback plan (because you know you’ll need it)**

* **Immediate**: set enableFlickWord = false.
* **Nuclear**: delete /features/flickword.html and the link vanishes into the night.
* **Modal mode**: set frame.src = '' on close; nothing persists.

**Summary of what you actually copy:**

* /features/flickword.html (the full game file from the canvas)
* The link button block (Step 2)
* The optional countdown script (Step 4)
* The optional modal block (Step 6) if you want iframe instead of a new tab
* The feature flag snippet (Step 5), because you enjoy sleeping at night

That’s it. You’ve got a **safe, external, link-driven** integration you can switch on/off without lighting your index on fire.