# Member Dashboard — v2.6 Approved ✓

**Status:** UI looks good based on your screenshot. Cards are clean, badges restored, favorite labeling fixed.

## How to (re)export — step‑by‑step (Windows / PowerShell)

1. Make sure the file in your repo is scripts/millions\_export\_member\_html.py (v2.6 from canvas).
2. Run:

* python scripts/millions\_export\_member\_html.py `  
   --season 2025 `  
   --week 1 `  
   --planner "picks/millions/millions\_planner.csv" `  
   --out "picks/millions/exports/member\_week\_1.html" `  
   --sort\_by kickoff

1. Open the HTML output in your browser.

## Visual QA checklist (1 min)

* Header has both teams and any **Top/Bottom 5 DVOA** pills.
* “Favorite (open/current)” shows a **team + spread** if determinable; otherwise row label becomes **Open / Current** with “HOME -x.x / AWAY +x.x”.
* **Total** shows a single number on the right (or **TBD**).
* DVOA rows (Total, Off, Def) appear for **both teams**.
* **Auto‑Notes** row appears **only** when there’s at least one chip.
* Kickoff shows local PT time.

## Planner columns the exporter expects (source-of-truth)

If a column is missing/blank, it simply shows TBD or hides the chip.

### Matchup & timing

* team, opponent, home\_or\_away (“HOME” / “AWAY”)
* kickoff\_local *(or)* kickoff\_pt *(optional)*
* \_kickoff\_sort\_key *(optional precalc)*

### Lines — spreads & totals

* open\_spread\_home, open\_spread\_away
* current\_spread\_home, current\_spread\_away
* circa\_line *(optional)*
* **Totals (any one works, first found wins):**
  + current\_total, market\_total\_current, closing\_total, open\_total, circa\_total

### Movement / notes

* line\_delta\_home *(pts; threshold default 2.5)*
* injuries\_key\_home, injuries\_key\_away *(short text OK)*
* weather\_notes *(free text)*

### Travel / rest & QB

* rest\_days\_home, rest\_days\_away, rest\_days\_diff *(home − away)*
* qb\_status\_home, qb\_status\_away *(e.g., OUT, Q, D, PROB, ACT)*

### Structured weather (optional numeric chips)

* wind\_mph, gust\_mph, temp\_f, precip\_code *(RAIN, SNOW, etc.)*

### Model & CLV (optional)

* edge\_spread\_team *(model vs market; ±)*
* clv\_delta *(closing − contest; + is good)*

### DVOA projections (used for rows + pills)

* Team side: team\_total\_dvoa\_proj, team\_off\_dvoa\_proj, team\_def\_dvoa\_proj
* Opp side: opp\_total\_dvoa\_proj, opp\_off\_dvoa\_proj, opp\_def\_dvoa\_proj

## Odds feed (fetch\_nfl\_odds) — mapping plan

**Goal:** generate/update the planner CSV with all fields above. If the function currently writes to a separate odds file, we’ll merge/join into the planner.

### What to pull (by market)

* **Point spread** (open + latest) for **home/away**.
* **Game total** (open + latest). If available, also **closing** when we backfill.
* **Book-specific**: Circa spread and total if offered.
* **Timestamps** so we can compute line\_delta\_home (current − open).

### How to compute/normalize

* Ensure **anti‑symmetry**: home\_spread = -away\_spread (fixes same-sign feeds).
* **Favorite detection**: favorite is the team with a **negative** spread.
* **Line move**: line\_delta\_home = current\_spread\_home − open\_spread\_home.
* **Totals precedence**: pick in this order ⇒ current\_total → market\_total\_current → closing\_total → open\_total → circa\_total.

### Output schema (merge into planner)

* open\_spread\_home, open\_spread\_away, current\_spread\_home, current\_spread\_away
* open\_total, current\_total, closing\_total, circa\_line, circa\_total
* line\_delta\_home
* Keep matchup keys: season, week, team, opponent, home\_or\_away, game\_id *(if you have one)*

### Sanity checks to add in code

* If only one side of spread is present → derive the other as its negative.
* If both sides present but not opposites → trust **home** and set away = -home.
* Drop obviously bad values (|spread| > 60, total < 20 or > 100).

## Tonight’s concrete steps

1. **Lock v2.6** (done ✅).
2. **Odds feed review**: open fetch\_nfl\_odds.py (or equivalent) and ensure it outputs the columns listed above.
3. **Join into planner** (by season/week + matchup keys).
4. Re-export the member HTML and skim the QA checklist.
5. If good, we’ll move to **Master Dashboard polish** next.

## If you want, I can wire this now

* Point me to the file (e.g., data/fetch\_nfl\_odds.py) and the planner build script (e.g., scripts/build\_planner.py).
* I’ll add the normalization + merge step and push a ready-to-run command.