AI Surf Report Builder

System Guide v1.0

Build your own surf reports.

A system guide for creating customized surf-report setups, long range reports, and validation records for a specific break or beach using your favorite AI.

https://surfreportbuilder.com

What This Is

AI Surf Report Builder is not a surf forecast website, app, or rating system.

It is a free DIY AI prompt kit.

You copy the main prompt in the kit, called the System Builder Prompt, and paste it into the AI of your choice. It asks a few setup questions, researches your selected surf break or beach, and creates reusable prompts customized to that location.

It's a prompt that builds you your own prompts.

You can use those new prompts to create Near Term / Session Reports, Long Range Reports, and optional validation records.

No coding skill is required for the basic setup. If you can copy, paste, and answer a few quick questions, you can use the quick-start version.

The goal is to give surfers a reusable way to inspect source data, translate it into local conditions, and keep their own records.

Website: https://surfreportbuilder.com

How It Works

In three steps:

Step 1. You copy the System Builder Prompt from the website and paste it into your AI.

Step 2. That prompt interviews you briefly — which break, when do you usually surf, how much do you want set up — then sends your AI out to research your specific spot. It finds the right ocean buoys, the right tide station, the right wind sources, the right marine forecasts. It checks which ones actually return usable data and discards the ones that don't. The result is a verified shortlist of sources for your exact break.

Step 3. It writes you new prompts using that shortlist — one for your next session, one for the upcoming week's swell outlook, optionally one for keeping a record of how reports compared to reality. From then on, you run those custom prompts whenever you want a report. You only go back to the System Builder Prompt if you want to set up another break or rebuild from scratch.

That's the whole shape. The rest of this guide explains the pieces and the optional extras.

Why This Works When Most AI Surf Prompts Don't

The honest reason most AI surf prompts produce mediocre reports is that AI tools are inconsistent at reading live ocean data. Buoy pages are JavaScript-heavy and often look blank to AI scrapers. Tide pages are worse. Wind data is hidden behind aviation databases that AI tools mostly can't reach reliably. Different AIs (Claude, ChatGPT, Perplexity) have different success rates at fetching this stuff, and even the same AI varies day to day.

This system solves that with a small piece of dedicated infrastructure called the Surf Report Worker. The Worker is a tiny program that lives on the internet, fetches buoy and tide data from official sources at the moment your AI asks for it, and hands back clean, reliable readings. Your AI doesn't have to scrape pages — it just asks the Worker "what's buoy 46221 doing right now?" and gets a current, parseable answer.

You don't need to do anything to use the Worker. The prompts the System Builder generates already know how to call it. The Worker is open-source, runs on free Cloudflare hosting, and is part of the public project.

One honest gap. The Worker currently handles waves and tide. Wind data is still fetched directly by your AI from public sources, which is more variable depending on which AI you use and what its web tooling can reach that day. This is the biggest remaining gap in the system and is the next thing being added to the Worker — so this section may already be out of date by the time you read it.

For the curious, the Worker source is at https://github.com/nfischbein/Surf-Report-Worker.

The Five Parts

The system has five named parts:

1. System Builder Prompt
2. Near Term / Session Report Prompt
3. Long Range Report Prompt
4. Validation Layer
5. Google Sheets Web App

1. System Builder Prompt

The starting prompt. This one is provided to you.

Find it on the website by clicking *Copy System Builder Prompt*. The website's copy window is the canonical source — it always reflects the latest version.

The System Builder Prompt builds all the other prompts. It's the multi-tool for surf-report prompt building.

It asks for:

1. the surf break or beach,

2. the session window,
3. Quick Start or Full System.

Then it builds the prompt or prompts for that break.

2. Near Term / Session Report Prompt

The near-term report prompt.

This prompt is created for you by the System Builder Prompt.

You run it when you want a Near Term Report for your next session, tomorrow morning, a specified session window, or a reconstructed past session.

It checks source freshness, reviews buoy, tide, wind, weather, and marine data where available, translates those readings into local break conditions, and can produce a structured record for future validation or storage.

Reports come in two layers: a plain-language section on top that reads like a friend texting you about conditions, and a technical appendix underneath with the raw data, source citations, and confidence reasoning. Most readers will only use the top section. The appendix is there if you want to dig in.

3. Long Range Report Prompt

The longer-horizon swell report prompt.

This is another output from the System Builder Prompt.

You run it when you want to look beyond the next session.

It looks for possible incoming swell events on roughly a 3–21 day horizon, with emphasis on the next 10 days. It separates confirmed incoming energy from developing signals, early signals, false alarms, and events unlikely to matter locally.

Same two-layer format as the Session Report: plain-language outlook on top, technical detail underneath.

4. Validation Layer

The optional post-session scoring layer.

The Validation Layer is optional. It works best as standing instructions inside an AI Project, Space, custom GPT, or saved chat.

You set it up once. After that, logging can be as short as: *"Today was 0," "Yesterday was -1, smaller than expected,"* or *"+2, much better than the report."*

If you are using a one-off chat, validation still works, but it is clunkier because you may need to paste the validation instructions or provide the session/report context again.

Google Sheets makes the layer useful over time by linking each short score to the original report, source data, and session ID.

5. Google Sheets Web App

The shared read/write layer for the Session Report Prompt, Long Range Report Prompt, and Validation Layer when users enable storage or validation.

It can store records from all three paths: Session Reports, Long Range Reports, and validation records, plus source-health notes, external observations, and analog matches.

Google Sheets is not required for Near Term / Session Reports or Long Range Reports, but it is strongly recommended to enable system improvement over time, especially when using the Validation function.

Quick Start

Quick Start gets you started generating your first surf report in just a few clicks.

Use this path if you want near-term reports for your break without setting up Google Sheets, validation, or Long Range Reports.

1. Click *Copy System Builder Prompt* on the website.
2. Paste it into your AI of choice.
3. Answer the setup questions.
4. Choose Quick Start.
5. Let the AI create your customized Near Term / Session Report Prompt.
6. Save that prompt and run it whenever you want a report.

Example runs:

> *Run my Session Report Prompt for tomorrow morning at 6:30 AM.*
>
> *Run my Session Report Prompt for today at 5:00 PM.*
>
> *Run my Session Report Prompt for Sunday morning and assume I prefer clean, smaller surf over bigger wind-swell.*

Full System

Full System creates the complete workflow.

Use this path if you want:

- Near Term / Session Reports,
- Long Range Reports,
- optional post-session validation,
- Google Sheets Web App storage,
- source-health tracking,

- comparison against prior sessions with similar buoy, wind, tide, and validation data.

1. Click *Copy System Builder Prompt* on the website.
2. Paste it into your AI of choice.
3. Answer the setup questions.
4. Choose Full System.
5. Choose Google Sheets Web App support if you want validation or durable storage.
6. Let the AI create the Near Term / Session Report Prompt, Long Range Report Prompt, Validation Instructions, and Sheets setup instructions.
7. Save the prompts and run them as needed.

Google Sheets is not required for Quick Start. Google Sheets is strongly recommended when you enable validation.

Which AI Should I Use?

This system depends on your AI being able to fetch current data from the web in real time. **If you're using an AI without web access, it won't work** — that's a hard floor, not a preference.

The system has been most thoroughly tested on **Claude.ai with web search enabled**, which is what we recommend for first-time setup. Other AIs with web browsing capability may work — your mileage may vary depending on the AI's web-fetching reliability.

One practical tip. If your first report comes back with "Speculative" confidence and several sources marked offline, try again in five minutes. Sometimes the system just needs a fresh fetch to clear up; an actual outage will still be there on a retry, but transient hiccups usually clear.

This section will be updated as the system is tested across more AIs. If you have results from a particular AI you'd like to share, the feedback link at the bottom of the website is the right place.

Source Policy

Reports primarily rely on measured data from public sources and extrapolations the AI performs from that data.

Reports do not use commercial surf-reporting forecasts, predictions, proprietary models, ratings, summaries, editorial calls, interpretations, hype, recommended boards, "best spots" calls, or commercial conclusions about expected surf quality.

Commercial surf-reporting sites can contribute only two categories of information:

1. **Real-time measured data**, such as live buoy readings, wind readings, water temperature, tide data, or other raw timestamped observations;
2. **Historical observed descriptions**, such as post-session notes, dated photos or videos, archived observed reports, or cam-observed notes that describe what happened after the fact.

Buoy and tide data is fetched through the Surf Report Worker (described earlier in this guide). Wind data is currently fetched by your AI directly from public aviation and weather services.

What This System Won't Do

Most surfers already know this, but it matters because the system may be shared outside its original context: this is not a lifeguard service, marine safety service, or hazard warning system. It may miss dangerous shorebreak, rip currents, lightning, water quality issues, closures, debris, access problems, or conditions that changed after the report was generated. **Check the beach and know your limits before entering the water.**

The system depends on the AI environment you use. Some AIs browse better than others. Some can read rendered pages reliably. Some lean too hard on raw XML or stale feeds. Some miss timestamps unless the prompt forces them to check.

The reports work best when the AI can:

- browse current web pages,
- read rendered buoy and weather pages,
- verify timestamps,
- compare multiple sources,
- explain missing or stale data,
- translate technical readings into plain surf terms.

What Your Reports Will Do

Your reports will:

- check a chain of relevant buoys instead of relying on one number;
- translate offshore buoy data into likely conditions at your specific break;
- explain how swell direction, period, tide, wind, bathymetry, headlands, islands, reefs, sandbars, jetties, and local blocking may change what actually reaches the beach;
- separate local, regional, upstream, and distant swell signals;
- distinguish groundswell from local wind sea;
- identify potential Long Range Report events without pretending they are guarantees;
- use measured data and observed descriptions, not commercial surf forecasts, ratings, models, or interpretations;
- explain stale, missing, or conflicting data instead of hiding it;
- confirm source timestamps before trusting readings;
- get better over time if you use Validation with Google Sheets for historical comparison and report refinement.

Reports come back in two layers — the plain-language section is what most readers want. The technical appendix below it is for when you want to verify the call yourself, see the source URLs, or feed the data into a record.

Customizing For Your Break

The System Builder Prompt starts with your break name, location, and session window.

It then researches the break and builds a local model. That model identifies:

- the most relevant local and regional buoys,
- upstream buoy chains,
- official tide and weather sources,
- common swell windows,
- known blocking features,
- likely wind sensitivities,
- tide windows,
- nearby headlands, islands, reefs, jetties, sandbars, or shelves that affect translation.

The generated prompt is not a generic surf prompt with your break name swapped in. It is a break-specific reporting prompt.

If the break has complicated local behavior, the prompt can include that complexity. If the break is straightforward, the prompt stays simpler.

Revising Your Report Prompt

Your first generated prompt is a starting version.

You can revise it whenever you like.

Useful revision requests:

> *Revise my Session Report Prompt to make the main report shorter but keep the data caveats.*
>
> *Revise my Session Report Prompt to be more skeptical of short-period wind-swell.*
>
> *Revise my Long Range Report Prompt to emphasize Southern Hemisphere swell events.*
>
> *Revise my Session Report Prompt to stop using go/no-go language and only describe conditions.*
>
> *Revise my Session Report Prompt to include a one-line beginner/intermediate/advanced suitability note.*

The Validation Layer gives the best revision material because it records where the report matched reality and where it missed.

Using The Near Term / Session Report Prompt

The Near Term / Session Report Prompt creates a report for a specific upcoming, current, or reconstructed session window.

It can run before the session, after the session, or for a specified future session.

Your Near Term / Session Report will not blindly repeat buoy numbers. It will translate the technical readings into plain surf terms. The plain-language layer at the top of the report explains what those readings likely mean at the break: face height, consistency, power, lulls, texture, shape, tide effect, and who the session is likely best for. The technical appendix below it cites the raw readings and sources.

Example translation:

> *The local buoy is showing 3 ft at 14 seconds from a direction that's partly blocked for this break. In plain terms, that probably means inconsistent waist-to-chest-high sets rather than a steady 3–4 ft surf day. Expect long waits between the better waves, with more power on the sets than the average size suggests.*

Another example:

> *The local buoy is showing short-period west wind-swell. That can make the buoy number look decent, but at the beach it usually means crumbly, uneven surf with short lines and less push. Expect more texture than power unless the wind backs off.*

Each report includes:

- an opening header with the break, date, session window, run timestamp, and overall confidence rating;
- a plain-language section called *The Call* — three to five sentences covering size, tide, wind, and who the session favors;
- a one-line *What Could Make This Wrong* — names a specific failure mode for the call rather than vague hedging;
- a short *Why Confidence Is High/Medium/Low/Speculative* — explains the rating in plain English;
- a technical appendix with source data, freshness states, conflict resolution if any, and a structured Source Health summary;
- an optional structured storage record when Google Sheets is enabled.

Using The Long Range Report Prompt

The Long Range Report Prompt looks beyond the next session.

It tracks potential swell events on a 3–21 day horizon, with special emphasis on the next 10 days.

It classifies signals as:

- confirmed incoming,
- likely developing,
- early signal,
- low-confidence signal,
- false alarm or low local relevance.

It compares source-region storm charts, official marine forecasts, upstream buoys, local confirmation buoys, and rendered model fields where available.

It searches for and compiles buoy signals farther away from your local region, then accounts for distance, likely travel time, swell period, direction, and local filtering before treating those signals as relevant to your break.

It uses travel-time logic cautiously. Longer-period swell travels faster than shorter-period swell, but arrival time also depends on storm duration, fetch movement, direction, dispersion, and local filtering.

Time-horizon rules:

- **3–5 days out:** size category only (small, moderate, large, etc.) — no specific face heights.
- **5–10 days out:** source region, direction window, possible arrival window, and confidence — no precise size.
- **10–21 days out:** early signal only — no surf-height prediction.

Like the Session Report, the Long Range Report is two layers. The top is a plain-language outlook with an *Events to Watch* list, a *What Could Make This Wrong*, and a confidence explanation. The technical appendix below it includes a structured event table, upstream basin-by-basin context, and source health.

Validation Layer

(Google Sheets recommended)

The Validation Layer is the optional post-session scoring and storage layer.

It records how closely the report matched reality and uses that feedback to calibrate future reports.

In plain terms: the system generates a report, you go surf, and afterward you tell the system how accurate the report felt. You give the session a score. The Validation Layer records that score with the original report and source data. Over time, those records help future reports compare new conditions against prior similar days and refine confidence.

Google Sheets is strongly recommended for Validation. Without Google Sheets, validation can help inside a single AI conversation, but it is not a dependable long-term record. With the Google Sheets Web App, each validation can be stored and linked to the original report by session ID.

The best setup is to save the Validation Instructions as standing instructions inside an AI Project, Space, custom GPT, or saved chat. Then logging a session can be as short as *"Today was 0"* or

"Yesterday was -1, smaller than expected."

Reality Score

The main validation score is Reality Score.

It answers one question: **was there a difference between the surf you experienced and the report?**

Use this scale:

- **-2** = reality was much worse than the report
- **-1** = reality was somewhat worse than the report
- **0** = reality matched the report
- **+1** = reality was somewhat better than the report
- **+2** = reality was much better than the report

A perfect match is 0. Better than the report is positive. Worse than the report is negative.

Example inputs:

> *Log validation for this morning: 0.*
>
> *Reality Score -1. It was smaller and softer than the report suggested.*
>
> *Reality Score +2. Much better than the report, cleaner sets and more power.*

Optional category scores (the system can work with only the main Reality Score):

- size_score
- shape_score
- wind_score
- tide_score
- consistency_score
- power_score
- usefulness_score

How Validation Improves Reports

The system can learn the more you use it. Google Sheets with validation makes that possible by preserving what the report predicted, what the source data showed, what actually happened, and how you scored it.

Future reports can compare new conditions against that history and adjust confidence when similar setups have overperformed or underperformed before.

Google Sheets Web App

Google Sheets is optional for basic reports. It is strongly recommended for validation.

A Google Sheets Web App lets your AI send structured records to your spreadsheet without giving the AI login access to your Google account.

In the full system, the Web App is the single shared read/write layer for the three report paths. Your Session Report Prompt can write near-term report records. Your Long Range Report Prompt can write swell-event and horizon records. Your Validation Layer can write post-session Reality Scores and notes. All three can read from the same archive later when the AI compares current conditions against prior reports, source snapshots, and validation history.

You set this up after the System Builder Prompt creates your custom prompts, and before you rely on validation history. The basic reports still work without it. The Web App matters when you want Session Reports, Long Range Reports, source snapshots, long-range events, and post-session scores stored in one place for future comparison.

The Web App can store:

- Session Report records,
- Long Range Report records,
- Long Range Report event records,
- buoy observations,
- tide observations,
- wind observations,
- source-health notes,
- Validation entries,
- external observed-condition notes,
- analog matches.

The spreadsheet becomes the system's durable memory.

Tabs (auto-created on first use):

1. sessions
2. long_range_reports
3. long_range_events
4. buoy_observations
5. tide_observations
6. wind_observations
7. validations
8. external_observations
9. analog_matches
10. source_health

You do **not** need to create these tabs by hand. The reference Apps Script creates each tab automatically the first time data is written to it.

The Web App uses a shared secret. Keep it private.

If you don't enable Google Sheets, the generated prompts can still produce useful reports, but the system loses reliable long-term validation memory.

Setting Up The Google Sheets Web App

Use this setup when you want validation and historical storage.

This guide uses a vetted reference implementation that you copy directly from the project repository — earlier versions had your AI generate the Apps Script code from scratch each time, which introduced variability.

You will need:

- a Google account,
- about ten minutes for first-time setup,
- a private place to store one short random string (your shared secret).

Step-by-step:

1. Create a new Google Sheet (or open an existing one you want to dedicate to surf reports). Give it a recognizable name like "Surf Report Archive." You do not need to add tabs — the script creates them.
2. Open the reference Apps Script file at this URL: `https://github.com/nfischbein/Surf-Report-Builder/blob/main/script-b/Code.gs`
3. On that page, click the "Raw" button. Select all of the file contents and copy them.
4. Back in your Google Sheet, open **Extensions** → **Apps Script**. A new browser tab opens with starter code in an editor.
5. Delete the starter code in the editor. Paste in the Code.gs you just copied.
6. Find the line near the top that reads: `const SHARED_SECRET = 'CHANGE_ME_TO_A_RANDOM_STRING';`
7. Change the string between the quotes to a random value of your own. Any 16+ character random string works. Keep it private — it is the only thing protecting your sheet. **Use your own value, not anything you've seen as an example anywhere else.**
8. Save the script (disk icon or Cmd/Ctrl+S).
9. Click **Deploy** → **New deployment**.
10. Click the gear icon next to "Select type" and choose **Web app**.
11. **Description:** anything you want, e.g. "Surf Report Builder v1."
12. **Execute as:** Me.
13. **Who has access:** Anyone.

"Anyone" sounds scary, but Google requires it so your AI can POST without logging in. The shared secret is what actually protects your data — anyone without it gets rejected.

1. Click **Deploy**. Authorize when prompted. The consent screen will warn that the app is unverified — that is expected because you wrote it yourself. Click **Advanced** → **Go to [project name]** → **Allow**.
2. Copy the Web App URL Google shows you after deployment. It looks like: `https://script.google.com/macros/s/AKfyc.../exec`
3. Test the deployment by visiting that URL in your browser. You should see a JSON response listing the accepted record types. If you see an error page or HTML, something went wrong with deployment.
4. Save your Web App URL and shared secret somewhere safe (a password manager works). You will paste both into your generated report prompts when those prompts ask for them.

After setup:

You have two values to paste into generated report prompts when those prompts ask for them:

- Your Web App URL.
- Your shared secret.

The Near Term / Session Report Prompt, Long Range Report Prompt, and Validation Layer each include placeholders for these values. Fill them in once after the System Builder Prompt generates each prompt.

If You Suspect Your Secret Has Leaked

1. Open your Apps Script editor again.
2. Change `SHARED_SECRET` to a new random string.
3. Save and redeploy: **Deploy** → **Manage deployments** → **pencil icon** → **New version** → **Deploy**.
4. Update the secret in every prompt that uses it.

The Web App URL itself does not need to change — just the secret.

Updating The Reference Code Later

If the project repository ships an update to Code.gs, you can refresh your deployment:

1. Open your Apps Script editor.
2. Replace the script contents with the updated Code.gs from the repository.
3. Restore your shared secret in the new code (the new file ships with the placeholder).
4. Save.
5. **Deploy** → **Manage deployments** → **pencil icon** → **New version** → **Deploy**.

Your existing data is preserved. Only the script behavior updates.

Safety notes: this script never reads from your sheet, only appends. It cannot delete or overwrite existing rows. It does not send any data anywhere except your own sheet.

Questions And Feedback

Use the website to provide feedback, report problems, or ask questions:

https://surfreportbuilder.com

Appendix A — Where To Find The Current Prompt

The current System Builder Prompt is always available at https://surfreportbuilder.com — copy from the website's copy window for best results. Copying from a PDF can introduce line-break or formatting errors.

The website always reflects the latest version. This guide describes how the system works; the website is the canonical source for the prompt itself.

For anyone who wants to see version history, contribute, or fork the project, the source is on GitHub at https://github.com/nfischbein/Surf-Report-Builder.