Tara web search and result navigation

One query, a local match, and four places where finding something is not the same as reaching it.

TARA · FIND A PRACTICE · CHAPTER 3 OF 5

Three things this chapter teaches, carried by one query.

RUNNING EXAMPLE

The query “calm”: seven local results, a synonym indicator, a row that matched without containing the word, and a click that records history and opens nothing.

LISTENING TIME

About 29 min · 22 slides

YOU WILL LEARN

1. **Everything here is local**
   Twenty-eight inline records, a local timer, and no request to any service.
2. **Order is two decisions, not one**
   A global score sorts, and then fixed type sections decide what is read first.
3. **A click is not a launch**
   History is written before the route decides, and meditations have no route at all.

By the end you can say what a result row has established, and what every step after it still owes.

What this screen actually searches

An inline inventory, a local timer, and three settled states a reader can confuse.

Section 1 of 4

1 What this screen actually searches
An inline inventory, a local timer, and three settled states that are easy to confuse.

2 How a match is made
Expansion, weighted substring scoring, fixed section order, and what the highlight does not show.

3 What the screen remembers
Spelling suggestions, the events that write history, its exact-string order and its durability.

4 Where a result goes
Identity mapping into routes, the selection boundary, the scoped interaction checks, and the close.

> Nothing on this screen asks a service anything, so nothing on it can fail the way a service fails.

One query, seven local results

The screen that runs the whole chapter, with the summary line that says how the match was made.

Implemented UI · source 1ceae52 · local example content



Query summary · local example content and expanded matching

1 The settled query
Matching runs on this value, 250 milliseconds after the last edit — not on what is being typed.

2 Clearing returns to the start
It empties the query and refocuses the input, and it leaves the recent list alone.

3 Seven results, and a synonym flag
The count is of local records with a positive score; the flag says the term list grew.

Three parts of the shipped summary, numbered in the order a reader should visit them.

> A found result and a supported destination are two separate outcomes, and this screen only produces the first.

Twenty-eight records, declared inline

The whole searchable pool is four constants in the component, counted here by kind.

Configured values · read from apps/oshun/web/src/components/domains/tara/TaraSearchEngine.tsx

12 records

Meditations

A duration, a category and authored activity counts.

6 records

Courses

Lesson counts and course metadata.

6 records

Teachers

Descriptions and follower counts.

4 records

Collections

Descriptions and item counts.

The whole searchable inventory, counted by kind. Ten trending queries sit in a second constant beside it.

> These counts describe source data; they say nothing about the size or the popularity of a live catalog.

The timer, and what Enter does not do

A later edit cancels the previous timer, and pressing Enter records a query without settling it.

A non-empty edit has arrived and a timer has been started or replaced.

The question the effect asks Has 250 ms passed since the last edit?
A new edit cancels the timer and starts another.

- yes → **Admitted** Settle and compute
  Copy the trimmed query, clear the searching flag, score the records.
- not yet → **Try again** Keep waiting
  A spinner and four skeleton rows, with no request anywhere.

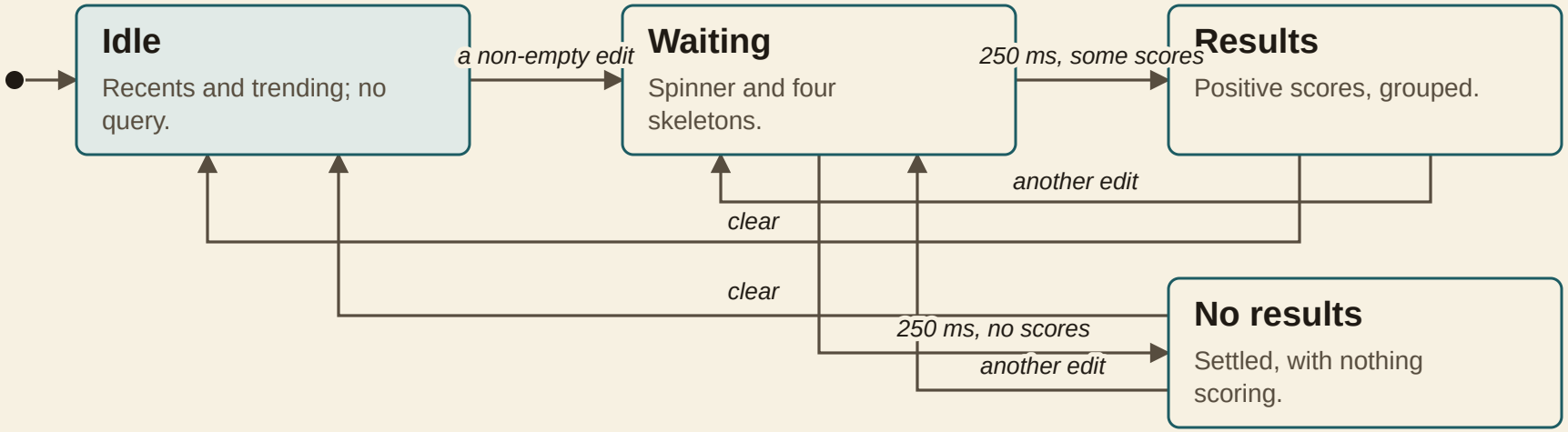
One question, re-asked whenever the input changes.

> Enter records the raw query; it does not make the results arrive any sooner.

Three settled states, and one of them is empty

Waiting, results and no results are different things, and none of them is a failure.

Local component states · pinned source



Every state this component has. There is no error state, because there is no request.

> Treat zero matches as a finished search; it is not loading, and it is not a network failure.

How a match is made

Expansion, a weighted substring score, a fixed section order, and a highlight that lags behind.

Section 2 of 4

1 What this screen actually searches
An inline inventory, a local timer, and three settled states that are easy to confuse.

2 How a match is made
Expansion, weighted substring scoring, fixed section order, and what the highlight does not show.

3 What the screen remembers
Spelling suggestions, the events that write history, its exact-string order and its durability.

4 Where a result goes
Identity mapping into routes, the selection boundary, the scoped interaction checks, and the close.

A result is in the list because of a term the member may never have typed.

One pass, and only the original words

A term added by expansion does not itself go looking for more terms.

TYPED: STRESS

One group, then stop

The anxiety group is visited; calm joins the term list.

WORKED OUTCOME

relax is not added, because calm was never typed

TYPED: CALM

Two groups at once

calm is both a member of anxiety and a key of its own.

WORKED OUTCOME

anxiety, stress and relax all join the list

EITHER WAY

Collected in one set

Repeated terms are collapsed before matching runs.

WORKED OUTCOME

each expanded term is matched once

Expansion visits the synonym groups of the typed words only, once each.

> The words the member typed decide which synonym groups are visited, and nothing else does.

Where the points come from

Four fields, four different weights, and popularity only after a non-zero text match.

Source field specimen · no member data

Declared score fixture · term calm

Text matches	`25 match points`	Title 10, subtitle 5, snippet 3, category 7.
Metadata boosts	`9 further points`	Rating 4, plus log of plays and log of followers.
Final score	`34, or 0 with no match`	Boosts are added only after a non-zero text total.

The substring cal produces the same result for this specimen, because the check is a substring test.

> A popular record with no text match scores zero, and the boosts never get a chance to run.

Sorted once, then re-ordered

A global score decides the ranking; four fixed sections decide the reading order.

	The global sort Every surviving record, in descending score.	The rendered order Four type sections, always in the same order.
What decides it	The weighted score, computed for this query.	The record’s type, fixed in the component.
The 12.35 teacher	Above the 7.8 course.	Below it, in the third section.
Within one section	Descending, as sorted.	Descending, as sorted.
A type with no matches	Contributes nothing.	Its section is omitted entirely.

> The page shows a type order, not a relevance order, so the ranking a reader infers from reading downwards is not the one the component computed.

One executed query, answered for the global sort and for the rendered order.

> A later section can hold a higher-scoring result than the last row of an earlier one.

Matched by a word that is not shown

This row is in the calm results, and the word calm appears nowhere in it.

Implemented UI · source 1ceae52 · local example content



One complete result row · local example content; no calm highlight

1 No calm anywhere

The word that was typed is not in the title, the subtitle, the snippet or the category.

2 relax and tension are

Both arrived through expansion, and both scored; the highlighter never sees them.

3 The arrow is not a route

It is part of the row; whether a click goes anywhere is decided three slides from here.

Three things this row shows, numbered in the order a reader should visit them.

> A missing highlight does not mean a low score; it means the highlighter was given different words.

What the screen remembers

A correction map, four events that write history, an exact-string order, and a write that can fail quietly.

Section 3 of 4

1 What this screen actually searches
An inline inventory, a local timer, and three settled states that are easy to confuse.

2 How a match is made
Expansion, weighted substring scoring, fixed section order, and what the highlight does not show.

3 What the screen remembers
Spelling suggestions, the events that write history, its exact-string order and its durability.

4 Where a result goes
Identity mapping into routes, the selection boundary, the scoped interaction checks, and the close.

The recent list records that a query was acted on, and nothing more than that.

One word, one fixed confidence

The suggestion corrects the first mapped word it finds, and it changes nothing until it is selected.

SLEEEP MEDITATOIN

Only the first mapped word

The helper stops at the first word it finds in the map.

RETURNED OR VISIBLE RESULT

sleep meditatoin — the second word is left as typed

BREATHWORK

Mapped to itself

The map contains self-mappings, and the helper does not skip them.

RETURNED OR VISIBLE RESULT

a suggestion whose text is the query

ON SELECTION

Applied, recorded, refocused

The proposed query is set, written to recents and the input regains focus.

RETURNED OR VISIBLE RESULT

the debounce then runs as it would for typing

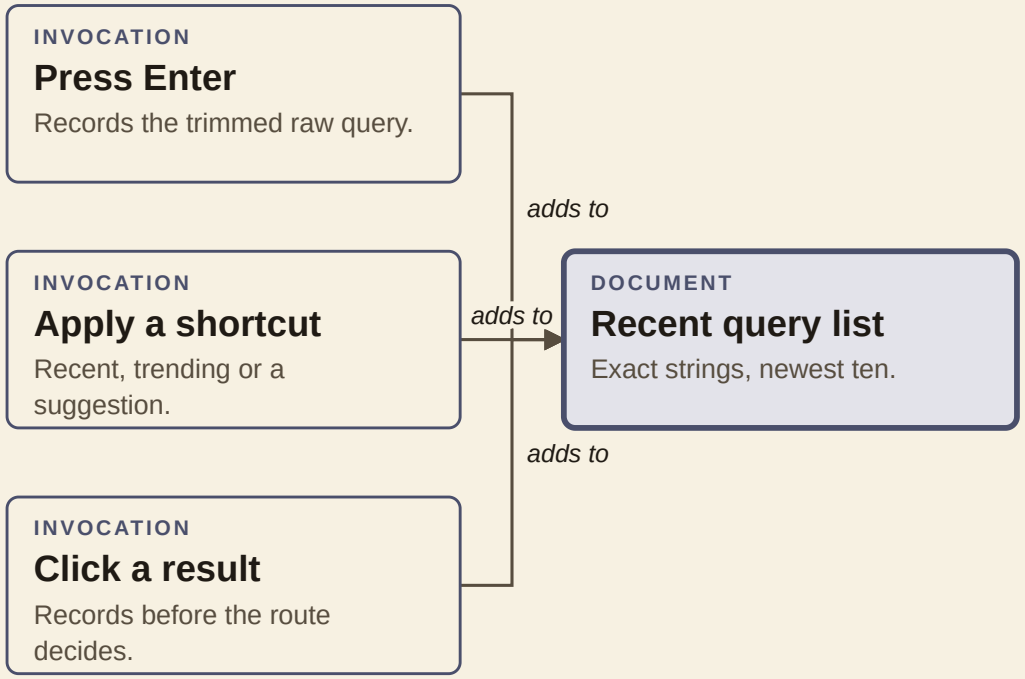
The correction map and its confidence are fixed values in the source.

> A suggestion can sit beside a full result list, and it can even suggest the query that was typed.

Four events write history; typing is not one

Enter, a query shortcut, a suggestion and a result click all record; editing records nothing.

Source interaction map · events that write history



Evidence

Three events write to the recent list; editing the input is deliberately not one of them.

> A recent entry proves an interaction with a query, and never proves that anything opened.

Ten exact strings, newest first

An existing exact match moves to the front, and a difference in casing is a different entry.

Source field specimen · no member data

Executed recent-history examples

Exact-string identity	`Sleep and sleep both survive`	The comparison is case sensitive.
Newest first, ten kept	`entry-11 … entry-2`	Twelve distinct additions leave the last ten.
Its own key	`tara-recent-searches`	Neither the query nor the list is written into the route.

Declared browser-fixture interactions, not a member’s history.

> A recent list is a bounded interaction history, not a normalized index of anything.

Visible now, and gone on the next mount

A failed write is swallowed, and what comes back is not checked for shape.

	What the screen shows The mounted component’s own state.	What the key holds The bytes a later mount will read.
Malformed JSON in the key	An empty recent list, with no error.	Whatever was there; parsing threw and was caught.
Valid JSON that is not an array	Whatever the object renders as; nothing checked it.	The object, returned unchanged by the helper.
A write that throws	The new chip, added to mounted state.	Nothing — the key is still absent.
After a remount with writes working	The chip is gone.	Still nothing to load.

> Only a remount answers the question a chip appears to answer, because every failure on this path is caught and discarded.

Three declared storage cases, answered for both columns.

> A chip on the screen does not prove that the next mount will be able to read it back.

Where a result goes

Identity mapping, a route that can return nothing, and what a scoped interaction check is worth.

Section 4 of 4

1 What this screen actually searches
An inline inventory, a local timer, and three settled states that are easy to confuse.

2 How a match is made
Expansion, weighted substring scoring, fixed section order, and what the highlight does not show.

3 What the screen remembers
Spelling suggestions, the events that write history, its exact-string order and its durability.

4 Where a result goes
Identity mapping into routes, the selection boundary, the scoped interaction checks, and the close.

The screen can find a thing it has no way of opening, and it does that most of the time.

Which identifiers become destinations

Three of the four result kinds can produce a route, and only for exactly-shaped identifiers.

COURSE

Exactly three digits

A prefix, a hyphen and three digits become the course route.

WHAT THE HELPER RETURNS

crs-001 → the Tara course path for 001

TEACHER

The same rule again

The teacher prefix, with the same three-digit requirement.

WHAT THE HELPER RETURNS

tch-001 → the Tara teacher path for 001

COLLECTION

An encoded parameter

The identifier is carried as a query parameter to the browser.

WHAT THE HELPER RETURNS

the browser resolves the selection itself

MEDITATION

No destination

The helper returns nothing, so the wrapper has nothing to push.

WHAT THE HELPER RETURNS

the click stays on the search page

All four kinds. Every supported route sits below the Tara domain path.

> An Open label on a row establishes a control, not a destination.

The click that records and stays

History is written first, and the route decision happens after it.

A result row has been clicked and the query has already been recorded.

The question the wrapper asks Did the helper return a destination?

The result kind and the shape of its identifier decide.

yes →

Admitted Request that route

A course, teacher or collection path — a request, not an arrival.

no →

Refused Stay on the search page

Meditation returns nothing, and the recent entry is already written.

One click. The recent entry is written before this question is asked.

> Separate the recorded selection from a launch; on this screen they are usually not the same event.

What the interaction checks cover

Focus, removal and reduced motion were observed exactly; everything around them was not.

FOCUS

Returned, not lost

Initial focus, applying a query and clearing all return it to the input.

WHAT WAS OBSERVED

clear search puts focus back in the field

REMOVAL

Stops at the chip

The remove control stops the click reaching the row behind it.

WHAT WAS OBSERVED

removing a chip does not apply its query

REDUCED MOTION

Animations disabled

The declared rows, highlights and skeletons drop their animation.

WHAT WAS OBSERVED

a result row reports an animation name of none

Observed behaviours of the pinned component, and the boundary around them.

> A scoped check is evidence about its scope, and accessibility acceptance is a larger thing.

Follow it all the way, or say where you stopped

Four different things were established, and a complete journey needs a fifth nobody has shown.

Illustrative case · executed source behaviour

“I found a practice — what happened next?”

Keep the query, the stored history and the destination independently inspectable.

Inspect the match

Its terms, its score and the section it was rendered into.

Inspect the saved query

Distinguish the mounted state from what a later mount can read.

Inspect the destination

Confirm the navigation, and the content it resolves on arrival.

Source-backed web behaviour; no live catalog, member or clinical acceptance.

> A complete search journey needs a destination and its content, and this screen ends before both.

Tara web search and result navigation

Three things to keep, and where the track goes next.

REMEMBER

1. **Everything here is local**
   A spinner is a timer, and an empty result is a finished search.
2. **Order is two decisions**
   A global score sorts; fixed type sections decide what is read first.
3. **A click is not a launch**
   History is written first, and most rows have no destination at all.

NEXT · CHAPTER 4 OF 5

Tara service discovery, saved items and launch

Discovery, saved items and launch: the paths into a practice that do not begin with a recommendation, and the destinations this screen could not reach — chapter four. →

Ask what a screen has established before letting it stand in for the journey around it.