Prompt to gamify Giuseppe

New Feature to add to Giuseppe Somm.

We will add gamifying features to Giuseppe in the style and tenure of DuoLingo

**Quizes**

The quizzes will be on a new page called “Learn Wine!” which will include the Giuseppe Avatar, and intro from Giuseppe about how the quizzes work, the badges the user can earn and why the user should learn more about wine.

Style of Quiz page

* Warm Mediterranean palette — terracotta reds, olive greens, vineyard golds.
* Soft cork-pop sound when scoring
* Light pour animation as the progress bar fills.

There will be both a “Pop-quiz” (5 questions in one minute) and a “sip and learn” (10 question, untimed quizzes) type that they choose when they click “Start Quiz” on the “Learn Wine!” page. The quiz will appear in the same page as a Quizlet style interface with thumbs up/down voting on each question enabled. Users will see the question and 4 optional answers as large buttons. They will click on the button to answer (no submit after – the button click is the submit).

When users answer correctly, have Giuseppe pop up in the center of the screen with a wink and say quiz\_messages.answer\_type=”correct” message, this will last 5 seconds or until the user click the “X” on the pop up to close it.

When wrong: have a Giuseppe image (to be defined) pop up in the center of the screen and say wine\_quiz.incorrect\_answer\_response message, this will last until the user click the “X” on the pop up to close it.

During the quiz, there will be a simple wine glass filling animation to indicate how many questions they got right.

**Study areas: Regions & Appellations, Grapes, Styles, Pairings, Classifications**

When the quiz is done, if the user has scored, on a Pop-quiz 80% = 4 or 5 correct or on a Sip and Learn quiz 90%= 9 or 10 correct, an image of Giuseppe with a raised glass of wine will appear mid-screen, duolingo style, with a quiz\_messages.answer\_type=”complete\_90” message. If they scored Pop-quiz 60% to 79% = 3 correct or for a Sip and Learn quiz 60% to 89% = 6,7 or 8 correct, the user will see a quiz\_messages.answer\_type=”complete\_70” message. If they scored less than 60% (Pop-quiz = 0, 1 or 2 correct; sip and learn = 0,1,2,3,4,5 correct) then use a quiz\_messages.answer\_type=”fail” message.

In the User profile, the system will store both all scores on all quizzes (**user\_study\_mastery table below)** AND the questions the user has already gotten right in any quiz (user\_quiz\_progress table; FK to wine\_quiz.question\_id) by wine\_quiz.study\_area.

A “Subject Mastery” page will show a gauge (like a cars speedometer) of progress towards Mastery of each study\_area. Users can NOT get credit for answering the same questions twice so all user correct answer tracking must include a question\_id to ensure no double counting. The gauge will go from 0 to 50 with five sections on the gauge in red (o to 9), pink (10 to 19), orange (20 to 29), yellow (30-39) and green (40 to 50).

Badge awards for quizes

| **Correct Answers** | **Gauge Color** | **Badge Name** | **Tone / Explanation** |
| --- | --- | --- | --- |
| **10–19** | 💗 Pink | Apprendista (Apprentice) | “You’re learning to swirl with confidence.” |
| **20–29** | 🧡 Orange | Degustatore (Taster) | “You can talk terroir and tannins like family.” |
| **30–39** | 💛 Yellow | Conoscitore (Connoisseur) | “Your palate has stories to tell.” |
| **40–49** | 💚 Green | Esperto (Expert) | “Giuseppe tips his hat. You are an expert of this subject.” |
| **50** | Lucide Crown | Maestro di Vino (Student becomes the Master) | “The Student becomes the Master!” |

New Table: wine\_quiz

| **Table** | **Column** | **Type** | **Notes / Constraints** |
| --- | --- | --- | --- |
| wine\_quiz | question\_id | bigint PK | Primary key (identity) |
|  | study\_area | text | e.g., “Regions & Appellations”, “Grapes”, “Styles”, “Pairings”, “Classifications” |
|  | question\_text | text | The Question |
|  | answer\_a | text | Multiple-choice A |
|  | answer\_b | text | Multiple-choice B |
|  | answer\_c | text | Multiple-choice C |
|  | answer\_d | text | Multiple-choice D |
|  | correct\_answer | char(1) | Must be A, B, or C (CHECK constraint) |
|  | incorrect\_answer\_response | text | Short friendly explanation shown when user is wrong |
|  | upvotes | integer | Default 0, non-negative |
|  | downvotes | integer | Default 0, non-negative |

New Table: quiz\_messages

| **Table** | **Column** | **Type** | **Notes / Constraints** |
| --- | --- | --- | --- |
| quiz\_messages | message\_id | bigint PK | Primary key (identity) |
|  | answer\_type | text | correct, complete\_90, complete\_70, fail |
|  | Message\_text | text | The answer shown for the answer\_type |

**New Table:** user\_quiz\_progress

| **column** | **type** | **description** |
| --- | --- | --- |
| id | bigint | PK |
| user\_id | uuid | FK → profiles.user\_id |
| question\_id | bigint | FK → wine\_quiz.id |
| study\_area | text | e.g. “Regions & Appellations” |
| is\_correct | boolean | was the answer correct |
| created\_at | timestamptz | timestamp |

**Unique constraint:** (user\_id, question\_id) — ensures no double counting.

**New Table: user\_study\_mastery**

| **column** | **type** | **description** |
| --- | --- | --- |
| user\_id | uuid | FK |
| study\_area | text | e.g. “Grapes” |
| correct\_unique\_count | int | 0–50 capped |
| badge\_tier | smallint | 0–5 (matches badge level) |
| updated\_at | timestamptz | auto |

Supabase function:

CREATE OR REPLACE FUNCTION update\_study\_mastery()

RETURNS trigger AS $$

BEGIN

IF NEW.is\_correct THEN

INSERT INTO user\_study\_mastery (user\_id, study\_area, correct\_unique\_count, badge\_tier, updated\_at)

VALUES (NEW.user\_id, NEW.study\_area, 1, 1, now())

ON CONFLICT (user\_id, study\_area)

DO UPDATE

SET correct\_unique\_count = LEAST(user\_study\_mastery.correct\_unique\_count + 1, 50),

updated\_at = now(),

badge\_tier = CASE

WHEN correct\_unique\_count + 1 >= 40 THEN 5

WHEN correct\_unique\_count + 1 >= 30 THEN 4

WHEN correct\_unique\_count + 1 >= 20 THEN 3

WHEN correct\_unique\_count + 1 >= 10 THEN 2

ELSE 1

END;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg\_update\_study\_mastery

AFTER INSERT ON user\_quiz\_progress

FOR EACH ROW EXECUTE FUNCTION update\_study\_mastery();

**Lucide Crown**

import { Crown } from "lucide-react";

export function MasteryBadge() {

return (

<div className="flex items-center gap-2 text-yellow-500">

<Crown className="h-6 w-6" strokeWidth={1.5} />

<span className="font-semibold">Maestro di Vino</span>

</div>

);

}

**Tasting Notebook**

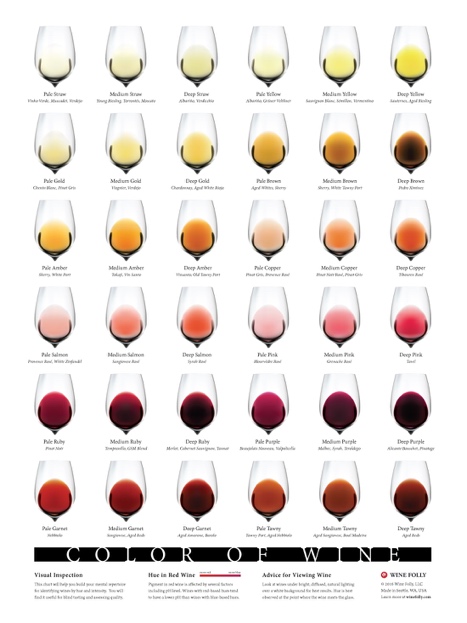
A new link off the Hamburger menu to “My Tasting Notebook”

The user can click into the Notebook to add a new tasting note or review past notes, including searching by name or rating. All wine details pages in My Cellar will also have an “Add Tasting Note” link

The user can also use the existing Label scanner (on the home page and in Add Wine) to scan a label and let the results fill in the form fields for the tasting note.

Each tasting note will include:

Date (auto populate TODAY(), Wine Name (text field), Producer (Text), Grape(s), Vintage (text), Alcohol % (text), Country (drop down countries\_regions.country\_name) that creates the list for the Region field (drop down countries\_regions.wine\_region) , Price, My Notes, Drink Starting, Drink By, bottle size, is Bubbly?   
  
It will also include more detailed capability for the core tasting note items: Appearance (color), Aroma, Palate and Rating (see below for details). Make these areas more substantial and noticable. They should have background that make it stand out .  
  
**Appearance** (add a toll tip with a popup including a wine color guide – images of wines of glass with a range of colors from clear to very dark and everything in between – see example but DO NOT USE as it is not mine)

Wine color Guide Example  


* Color (Drop down with colors that match the Wine Color Guide names listed in order from Lightest to darkest)

**Aroma** – including at 30 checkboxes of typical terms used to describe the aroma of wines that are sufficient to describe any type of wine from anywhere

**Palate –** use radio buttons

* Sweetness 1 to 10 sliding scale identical to that used on /profile page. Match to profile\_taste\_preferences.sweetness) with 1 being Extra Dry and 10 being Sweet as can be
* Acidity 1 to 10 slider (like on /profile page and in profile\_taste\_preferences.acidity) with 1 being No acidity and 10 being High acidity
* Body 1 to 10 slider (like on /profile page and in profile\_taste\_preferences.body) with 1 being Light and 10 slider being Bold
* Tannins 1 to 10 slider (like on /profile page and in profile\_taste\_preferences.tannin) with 1 being Easy Drinking and 10 being Makes me pucker
* Oak 1 to 10 slider (like on /profile page and in profile\_taste\_preferences.oak) with 1 being No oak and 10 being Bit an oak tree
* Old World Bias 1 to 10 slider (like on /profile page and in profile\_taste\_preferences.old\_world\_bias) with 1 being The New World and 10 being Old World Favorite
* Finish 1 to 10 slider with 1 being short as a cork and 10 being as long as a vineyard

**Rating**: a 5 wine bottle scoring scale. Use the same rating that already exists on the wine details in My cellar.

Also wire into the Badge System for the Tasting Note Virtuoso badge

All Tasting notes will be saved as a page in a tasting notebook and will be searchable.

At the end of every tasting note, the user will be prompted to ask “Add to cellar?” and if they say Yes, prompt for “How many bottles?” and allow them to enter a number between 1 and 999. If they are adding to cellar, copy all fields collected that match the cellar\_items table into that table (schema below).

create table public.cellar\_items (

bottle\_id bigserial not null,

user\_id uuid not null,

wine\_id bigint not null,

quantity integer not null default 1,

where\_stored text null,

value numeric(12, 2) null,

status text not null default 'stored'::text,

created\_at timestamp with time zone not null default now(),

updated\_at timestamp with time zone not null default now(),

currency text not null default 'USD'::text,

my\_notes text null,

my\_rating integer null,

drink\_starting character varying null,

drink\_by character varying null,

typical\_price numeric(10, 2) null,

ratings text null,

color text null,

alcohol numeric(5, 2) null,

bottle\_size text null,

bubbly text null default 'No'::text,

constraint cellar\_items\_pkey primary key (bottle\_id),

constraint cellar\_items\_user\_id\_wine\_id\_key unique (user\_id, wine\_id),

constraint cellar\_items\_user\_id\_fkey foreign KEY (user\_id) references profiles (user\_id) on delete CASCADE,

constraint cellar\_items\_wine\_id\_fkey foreign KEY (wine\_id) references wines (wine\_id) on delete CASCADE,

constraint cellar\_items\_status\_check check (

(

status = any (

array['stored'::text, 'drank'::text, 'lost'::text]

)

)

),

constraint cellar\_items\_my\_rating\_check check (

(

(my\_rating >= 1)

and (my\_rating <= 5)

)

),

constraint cellar\_items\_quantity\_check check ((quantity >= 0)),

constraint cellar\_items\_bubbly\_check check (

(

bubbly = any (array['Yes'::text, 'Slight'::text, 'No'::text])

)

)

) TABLESPACE pg\_default;

create index IF not exists idx\_cellar\_user\_status on public.cellar\_items using btree (user\_id, status) TABLESPACE pg\_default;

create index IF not exists idx\_cellar\_wine\_id on public.cellar\_items using btree (wine\_id) TABLESPACE pg\_default;

create index IF not exists idx\_cellar\_user\_id on public.cellar\_items using btree (user\_id) TABLESPACE pg\_default;

create trigger trg\_cellar\_items\_touch BEFORE

update on cellar\_items for EACH row

execute FUNCTION tg\_touch\_updated\_at ();

**Giuseppe Somm Tests**

**Badges**

Users will be able to earn Badges which will show up on the My Awards page (already implemented with Quiz scores). Follow the design used in my other application RecipeChef /users/josephvaleri/recipe-chef/src/app/badges which can be seen at: https://recipe-chef-omega.vercel.app/badges

| **Badge Name** | **Category** | | **Icon Idea** | **Tier Thresholds** | | **Description** | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Giuseppe Somm** | Scholar | | 📚🍷 | 5 / 15 / 30 / 60 / 100 quizzes completed | | Completed Somm training modules | |
| **Label Whisperer** | Collector | | 🏷️ | 10 / 25 / 50 / 100 / 200 scans | | Scanned and identified wine labels | |
| **World Cellar Traveler** | Explorer | | 🌍🍇 | 5 / 10 / 25 / 50 / 100 countries | | Logged wines from multiple countries | |
| **Tasting Note Virtuoso** | Taster | | 📝🍷 | 10 / 25 / 50 / 100 / 250 notes | | Logged detailed tasting notes | |
| **Student Becomes the Master** | Scholar | |  | 10,20,30,40,50 | | Mastery or any study\_area in the Quiz section | |
| **Giuseppe’s Favorite Student** | Scholar | | 🧠🍇 | 10 / 30 / 60 / 120 / 200 correct answers | | High quiz performance | |
| **Epicurean Explorer** | Food Pairing | | 🍽️ | 5 / 15 / 30 / 60 / 100 pairings logged | | Frequent food pairing use | |
| **The Archivist** | Collector | | 🏺 | 25 / 50 / 100 / 200 / 500 wines added | | Expanded personal cellar | |
| **The Bard of Bottles** | Storyteller | | 🎤🍾 | 5 / 20 / 50 / 100 / 250 stories shared | | Shared reviews or posts | |
| **The Alchemist** | Artisan | | ⚗️🍷 | 10 / 25 / 50 / 100 / 200 analysis | | Used AI label analyzer / vintage evaluator | |
| |  | | --- | | **La Prima Bottiglia** | |  | |  | |  | |  | | --- | | First wine scanned | | | |  | | --- | | “The journey begins…” | | |
| **Name** | | **Trigger** | | | **Reward Vibe** | |
| **La Prima Bottiglia** | | First wine scanned | | | “The journey begins…” | |
| **Wine Time Traveler** | | Added wine older than 1980 | | | “Respect for the ancients.” | |
| **The Polyglot Palate** | | Logged wines from 10+ countries | | | “You taste the world.” | |

**1.1. Taste Profile Builder**  
Let users calibrate their palate: ask them to rate a few wines or flavors (fruit, oak, tannin, acidity, sweetness). Pick wines they like  
→ Output a “Giuseppe Taste Signature” and personalized wine recommendations.

**6. Education & Gamification**

**6.2. Wine Quiz Challenges**  
Offer difficulty tiers and social leaderboards.  
Connect to cellars: quiz on their own bottles.

**6.3. “Flavor Wheel Explorer”**  
Interactive visual of aromas and flavor families; click → see wines that express them.

**2.4. Verified Sommelier Program**  
Allow wine pros to earn a “Verified Sommelier” badge and publish short “Sommelier Picks.”  
→ Adds authority and content generation.

**5.2. Wine Experiences Near Me**  
Geo-based recommendations: tastings, wine bars, vineyards, festivals.

Future

**7.3. Integration with RecipeChef / Chef Tony**  
Cross-recommend: “This wine goes with your saved recipe.”  
→ Shared taste graph between your apps.

**7.4. Voice Mode**  
“Hey Giuseppe, find a bottle for dinner tonight.”  
Voice input + audio response = accessibility + novelty.

**3.3. Winery / Merchant Portal**  
Let producers create profiles, upload tasting notes, vintage data, and events.  
You moderate or verify them.

**5.1. Wine Travel Planner**  
Integrate with Google Maps + your region DB:

“Plan a 3-day Sangiovese road trip in Tuscany.”  
→ Giuseppe builds a route of wineries, restaurants, and hotels.

**5.2. Wine Experiences Near Me**  
Geo-based recommendations: tastings, wine bars, vineyards, festivals.

**5.3. Integration with Tenuta Valeri & Partner Estates**  
Feature special tastings, booking links, and immersive “Meet the Winemaker” sessions.  
(You can later monetize via hosted tours.)