

Opik Integration in FretCoach

Workspace: padmanabhan-r-7119 Projects: FretCoach | FretCoach-Hub Link: [View Workspace →](#)

Overview

FretCoach traces three distinct AI coaching features — real-time live coaching, AI practice recommendations, and a web based chatbot agent (AI Coach). Opik provides the observability layer to monitor, evaluate, and continuously improve these features in production, giving us visibility into prompt quality, token costs, response latency, and AI coaching quality at scale.

Features Implemented

1. Traces with Metadata and Tags

All LLM calls are logged as traces in Opik with structured tags for filtering, organization and applying targeted online evaluation rules.

Hub Coach Chats:

- Tags: `ai-coach-chat`, `fretcoach-hub`, `from-hub-dashboard`, `gemini-2.5-flash`, `practice-plan`
- Tracks AI coach conversations in the web dashboard

The screenshot shows the Opik dashboard interface. On the left, there's a sidebar with navigation links like Home, Dashboards, Observability, Projects, Experiments, Datasets, Annotation queue, Prompt engineering, Prompt library, Playground, Optimization, Optimization study, Production, Online evaluation, Alerts, Star (17.5K), Configuration, Support hub, and Invite a teammate. The main content area has a header "Trace - 18 spans". It lists several spans: LangGraph (14.6s), agent (1.6s), tools (0.7s), ChatGoogleGenerativeAI (0.8s), execute_sql_query (0.7s), agent (0.8s), and ChatGoogleGenerativeAI (0.8s). Each span has a timestamp, count, percentage, and cost. To the right, there's a "LangGraph" diagram with nodes for start, end, agent, and tools, and arrows indicating flow. Below the diagram, there's a "Feedback scores" section with a table showing scores for Hub Response Clarity, Hub Answer Correctness, Hub Actionability, Hub Data Groundedness, and Hub Context Usage Quality, all rated as 1 with positive comments. The URL in the browser bar is comet.com/opik/padmanabhan-r-7119/projects/019bfd9b-526d-74f8-9c68-94743589a0b1/traces?time_range=alltime&size=100&height=small&traces_filters....

Hub coach chat traces with proper tags in Opik dashboard

AI Mode (Practice Recommendations):

- Tags: `fretcoach-core`, `gpt-4o-mini`, `ai-mode`, `fretcoach-studio`, `practice-recommendation`

- Tracks personalized practice recommendations

The screenshot shows the Comet Debugger interface with the URL `comet.com/opik/padmanabhan-r-7119/projects/019bcefc-a27c-718d-8c5f-36472d5decb2/traces?time_range=past7days&size=100&height=small&traces_filters=...`. The left sidebar has sections like Home, History, Observers, Promises, Plugins, Optimizations, Productivity, and Statistics. The main pane displays a trace for a 'RunnableSequence' with a duration of 3.1s, 2084 events, and a cost of \$0.001. It includes details about ChatOpenAI and RunnableLambda. On the right, there's a 'Details' tab showing a JSON response from the AI mode, which includes a 'PRACTICE HISTORY' section with session statistics and a 'RECENTLY PRACTICED SCALES' section with a single entry for 'A Minor' pentatonic scale.

```

{
  "trace": {
    "id": "RunnableSequence",
    "start_time": "2024-07-18T12:41:00Z",
    "end_time": "2024-07-18T12:41:03.100Z",
    "duration": "3.1s",
    "events": 2084,
    "cost": "$0.001"
  },
  "details": [
    {
      "label": "PRACTICE HISTORY",
      "value": [
        "Total sessions: 8",
        "Average pitch accuracy: 71.6%",
        "Average scale conformity: 21.1%",
        "Average timing stability: 56.5%",
        "Weakest area: scale"
      ]
    },
    {
      "label": "RECENTLY PRACTICED SCALES",
      "value": [
        "[{"scale_name": "A Minor", "scale_type": "pentatonic", "times_practiced": 8, "avg_pitch": 0.71617078055838, "avg_scale": 0.210802486501322, "avg_timing": 0.565457327091952, "last_practiced": "2026-01-30T23:11:58.958641"}]"
      ]
    }
  ]
}

```

AI mode practice recommendation traces

Live AI Feedback in Session:

- Tags: `fretcoach-core`, `gpt-4o-mini`, `ai-mode`, `fretcoach-studio`, `live-feedback`
- Tracks real-time coaching feedback during practice
- TTS audio generation traced separately via `@track` decorator for independent failure tracking and latency monitoring

The screenshot shows the Comet AI interface for monitoring and debugging AI applications. A specific trace is selected, labeled "ChatOpenAI", which consists of 1 span. The trace details provide performance metrics: duration of 2s, 389 events, and a cost of <\$0.001. The provider for this trace is identified as "gpt-4o-mini". The interface allows users to view detailed logs, feedback scores, input prompts, generated outputs, and metadata in YAML format.

Live AI feedback traces during practice sessions

2. Thread Management

Structured `thread_id` to group related LLM calls and maintain conversation context.

Thread Naming Conventions:

- **Hub Coach:** `hub-aicoach-chat-{timestamp}` - Groups all coach chat messages for a user
- **AI Mode:** `{deployment}-ai-mode-{practice_id}` - Maintains thread across recommendations
- **Live Feedback:** `{session_id}-live-aicoach-feedback` - Groups feedback within a practice session

A screenshot of the Opik interface showing a list of thread IDs. Each entry includes a timestamp, trace ID, message content, and JSON representation of the messages. The interface has a search bar at the top and various filters and sorting options.

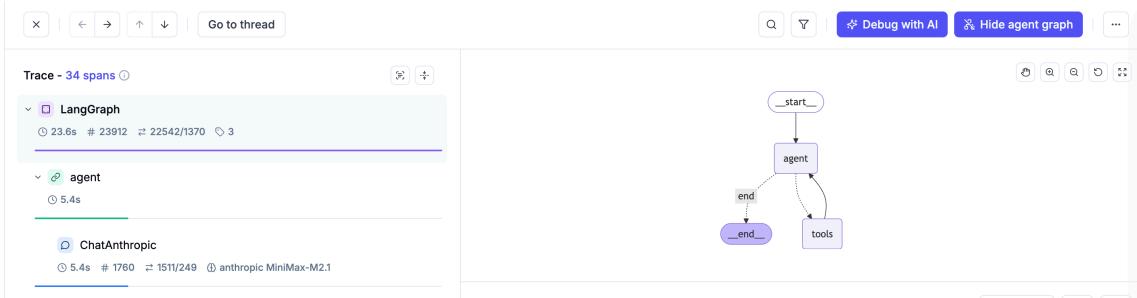
ID	First message	Last message
hub-aicoach-chat-1770250552190	How am I doing compared to average?	{"messages": [{"content": "Hi! I'm your AI practice coach."}]}
hub-aicoach-chat-1770223524432	How am I doing compared to average? And what day is it?	{"messages": [{"content": "Hi! I'm your AI practice coach."}]}
hub-aicoach-chat-1770217604585	How am I doing compared to average?	{"messages": [{"content": "Hi! I'm your AI practice coach."}]}
hub-aicoach-chat-1770215363136	What should I practice today?	```json { "focus_area": "Scale Conformity", "current_level": "Beginner", "target_level": "Intermediate" }```
hub-aicoach-chat-1770215206179	What should I practice today?	{"messages": [{"content": "Hi! I'm your AI practice coach."}]}
e868ca6e-c204-45e8-83c0-0be930248505-live-aicoach-feedback	Enabled metrics: Pitch Accuracy, Scale Conformity, Timing	Timing is spot on at 100%, but scale conformity is low.
6db7f5e0-1bdd-484d-9b1b-51059b3f8e90-live-aicoach-feedback	Enabled metrics: Pitch Accuracy, Scale Conformity, Timing	Timing is excellent at 97%, but scale conformity is low.
0a7d4419-a110-4ff7-a8b6-ae56363283d7-live-aicoach-feedback	Enabled metrics: Pitch Accuracy, Scale Conformity, Timing	Timing is excellent at 98%, but scale conformity is low.
hub-aicoach-chat-1770035493514	What should I practice today?	{"messages": [{"content": "Hi! I'm your AI practice coach."}]}
hub-aicoach-chat-1770035435207	What should I practice today?	{"messages": [{"content": "Hi! I'm your AI practice coach."}]}
hub-aicoach-chat-1770035366500	What should I practice today?	{"messages": [{"content": "Hi! I'm your AI practice coach."}]}
hub-aicoach-chat-1770035155059	Show me my progress	{"messages": [{"content": "Hi! I'm your AI practice coach."}]}

Thread IDs grouping related traces in Opik

3. Agent Graph Visualization

LangGraph execution flows visualized in Opik using `workflow.get_graph(xray=True)`.

Shows complete agent reasoning path: agent → tool calls (`execute_sql_query` , `get_database_schema`) → decision nodes → response.



LangGraph agent execution flow in Opik

4. Annotation Queues

Used annotation queues for human-in-the-loop evaluation of agent outputs.

Implementation:

- Custom feedback definitions for LLM output quality
- Manual review and rating using custom criteria
- Structured feedback collection for agent improvements

The screenshot shows the 'AI Coach Chat Thread Quality Checks' section of the Comet platform. It displays a list of annotation queue items. Each item includes the thread ID, the first message, the last message, and a comment from a user. The comments highlight specific issues such as 'Very bad, no practice plan actually displayed.' and 'Add guard rails, as it is responding to random texts.'

ID	First message	Last message	Comments
thread-1769613376415	What should I practice today? I feel like doing some GM Minor. Can you make me a practice plan?	{"messages": [{"content": "Hi! I'm your AI practice coach. I can help you analyze your performance, suggest exercises, and answer questions about your practice"}]	Very bad, no practice plan actually displayed.
thread-1769599262489	Show me my progress	{"messages": [{"content": "Hi! I'm your AI practice coach. I can help you analyze your performance, suggest exercises, and answer questions about your practice"}]	Add guard rails, as it is responding to random texts
thread-1769592723853	show me the scales that i have showed most improvement in	{"messages": [{"content": "Hi! I'm your AI practice coach. I can help you analyze your performance, suggest exercises, and answer questions about your practice"}]	Not many details
thread-1769592573338	show me the scales that i have showed most improvement in	{"messages": [{"content": "Hi! I'm your AI practice coach. I can help you analyze your performance, suggest exercises, and answer questions about your practice"}]	-
thread-1769591135023	Show me my progress	{"messages": [{"content": "Hi! I'm your AI practice coach. I can help you analyze your performance, suggest exercises, and answer questions about your practice"}]	-

Annotation queue with reviewed LLM outputs

5. Datasets and Prompts

Created datasets and prompts for reproducible experiments and evaluations.

- Curated test cases from real user sessions
- Used in experiment runs for testing
- Version-controlled coaching prompt templates

The screenshot shows the 'Datasets' section of the Comet platform. It displays a list of datasets. Each dataset entry includes the name, description, number of items, and creation date. The datasets listed are 'fretCoach_live_ei_feedback_val_9', 'fretcoach_live_ai_feedback_train_25', 'FretCoach Live AI Feedback', and 'FretCoach Hub AI Coach Chat'.

Name	Description	# Item co...	Most recent exper...	Created
fretCoach_live_ei_feedback_val_9		9	02/03/26 05:06 PM	02/03/26 04:11 PM
fretcoach_live_ai_feedback_train_25		25	02/03/26 04:48 PM	02/03/26 04:11 PM
FretCoach Live AI Feedback		44	02/03/26 03:48 PM	...
FretCoach Hub AI Coach Chat		10	01/27/26 10:24 AM	...

Datasets created for experiment runs

The screenshot shows a web browser window for the URL comet.com/opik/padmanabhan-r-7119/prompts/019c234c-a082-7421-bcb0-81853b7dd0ba?tab=prompt. The page title is "Live AI Feedback - System Prompt". The sidebar on the left includes icons for Home, AI, Study, Blogs, Github, Great Learning, and Competitor Proj...". The main content area has tabs for "Prompt", "Experiments", and "Commits". A button "Edit prompt" is visible. The "Prompt" tab is active, showing a text input field with placeholder "Text prompt" and a "Raw view" link. The text in the prompt is as follows:

You are a direct guitar coach giving quick real-time feedback. Your feedback MUST be 1-2 sentences, maximum 30 words total.
Format: "[What's good], but [what's weak] - [specific actionable fix based on performance context]"
To improve insight relevance, always relate feedback to specific performance scores, especially scores above 0.700. Include contextual details from the player's playing style to inform suggestions:

- Pitch Accuracy: How cleanly notes are fretted (low = finger pressure issues)
→ Fix: "ease finger pressure to improve note clarity" or "focus on clean fretting by adjusting finger placement"
- Scale Conformity: Playing correct scale notes across fretboard positions (low = stuck in one position or wrong notes)
→ Fix: "explore positions 5-7 to enhance versatility" or "move up the fretboard to discover new notes"
- Timing Stability: Consistency of note spacing (low = rushing, dragging, uneven rhythm)
→ Fix: "use a metronome at 60 BPM to develop timing" or "slow down and count to create consistent spacing"

Be direct and conversational, and vary your wording. Ensure your suggestions are anchored in the player's specific

Saved prompts

6. Experiments and Custom Metrics

Evaluated LLM performance using both default Opik metrics and custom-created metrics.

Default Metrics:

- Opik's built-in evaluation metrics for response quality

Custom Metrics:

- Domain-specific metrics tailored to guitar coaching context
- Measures coaching quality and relevance

comet.com/opik/padmanabhan-r-7119/experiments/019c2318-01d8-71a7-b2f7-c796b0577a6b/compare?experiments=%5B%019c2348-ef3f-7b71-97c5-88c73017e... ☆ 🔍

padmanabhan-r-7119 / Experiments / fretcoach-default-metrics-eval

fretcoach-default-metrics-eval

02/03/26 05:05 PM | fretCoach_live_ai_feedback_val_9 | Traces

Metrics: answer_relevance_metric (avg) 0.9044444444444444, context_precision_metric (avg) 0.7888888888888889, context_recall_metric (avg) 0.8388888888888889, hallucination_metric (avg) 0, levenshtein_r

Experiment items Configuration Feedback scores

Experiment items are individual evaluations that connect a dataset sample with its LLM output, feedback scores, and trace. Read more

Search dataset items Compare

ID (Dataset item)	Dataset	input	Evaluation task	input	output	reference	Duration	Total tokens
019c2325-43c2-73b...	Your pitch accuracy is strong, but timing stability is lacking. Slow	Enabled metrics: Pitch Accuracy, Timing Stability Pitch 89%, Timing	0: "You are a direct guitar coach giving	Enabled metrics: Great pitch accu...	Your pitch accur...	68.4s	p50 67.6s avg 13042.667	12,70
019c2325-43c2-73b...	Timing is solid, but scale conformity is weak. Focus on exploring different	Enabled metrics: Pitch Accuracy, Scale Conformity, Timing Stability	0: "You are a direct guitar coach giving	Enabled metrics: Great timing stab...	Timing is solid, b...	70.3s	14,86	
019c2325-43c2-73b...	Excellent pitch accuracy, but timing stability could improve.	Enabled metrics: Pitch Accuracy, Timing Stability Pitch 97%, Timing	0: "You are a direct guitar coach giving	Enabled metrics: Great job on pitc...	Excellent pitch a...	71.5s	13,56	

Experiments with default Opik metrics

comet.com/opik/padmanabhan-r-7119/experiments/019c2318-01d8-71a7-b2f7-c796b0577a6b/compare?experiments=%5B%019c2345-2271-71cd-a540-ab7b48b9... ☆ 🔍

padmanabhan-r-7119 / Experiments / fretcoach-coaching-feedback-eval

fretcoach-coaching-feedback-eval

02/03/26 05:01 PM | fretCoach_live_ai_feedback_val_9 | Traces

Metrics: coaching_quality (avg) 0.7888888888888889

Experiment items Configuration Feedback scores

Experiment items are individual evaluations that connect a dataset sample with its LLM output, feedback scores, and trace. Read more

Search dataset items Compare

ID (Dataset item)	Dataset	input	Evaluation task	input	output	reference	Duration	Total tokens	Estimated cost
019c2325-43c2-73b...	Your pitch accuracy is strong, but timing stability is lacking. Slow	Enabled metrics: Pitch Accuracy, Timing Stability Pitch 89%, Timing	Enabled metrics: Great pitch accu...	Your pitch accur...	4.1s	p50 3.1s avg 0	0		
019c2325-43c2-73b...	Timing is solid, but scale conformity is weak. Focus on exploring different	Enabled metrics: Pitch Accuracy, Scale Conformity, Timing Stability	Enabled metrics: Great timing stab...	Timing is solid, b...	3.1s	-	-		
019c2325-43c2-73b...	Excellent pitch accuracy, but timing stability could improve.	Enabled metrics: Pitch Accuracy, Timing Stability Pitch 97%, Timing	Enabled metrics: Great job on pitc...	Excellent pitch a...	3.2s	-	-		
019c2325-43c2-73b...	Timing is decent,	Enabled metrics:	Enabled metrics: Good timing, but ...	Timing is decent,...	3.5s	-	-		

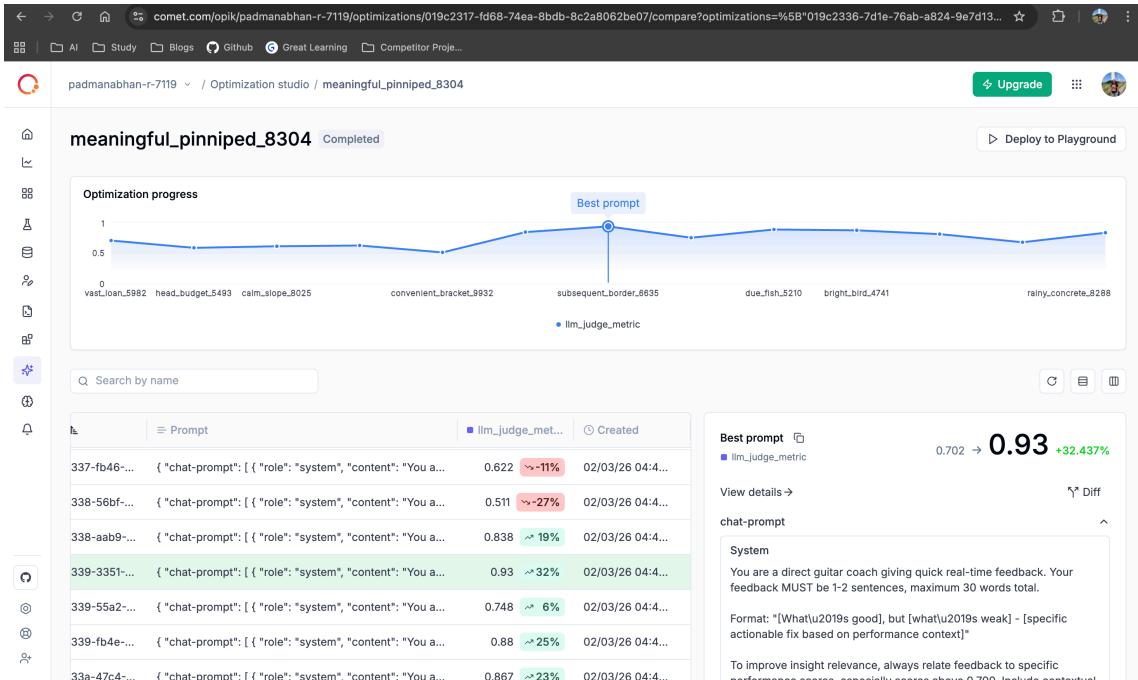
Experiments with custom metric

7. Optimization Studio

Used Optimization Studio with HRPO (Hierarchical Reflective Prompt Optimizer) to improve the prompt used in the live feedback module.

Results:

- 32% increase in `llm_judge_metric` custom metric
- Improved coaching feedback quality
- Optimized for better real-time guidance



Optimization Studio results for live feedback prompt

8. OpikAssist for Token Usage Optimization

Used OpikAssist to analyze traces and optimize token usage for hub coach chats.

Problem Identified:

- Excessive token usage (4,877 tokens) and long duration (7,873 ms)
- Lengthy prompts with redundant context and full SQL data

Actions Taken:

- Refined system and user prompts based on OpikAssist suggestions
- Streamlined SQL result formatting to essential data only
- Removed redundant context and consolidated guidelines

Results:

- Significant token usage reduction
- Improved response latency
- Better cost-performance ratio

The screenshot shows the Opik platform's interface. On the left, a sidebar navigation includes Home, Dashboards, Observability, Projects (selected), Evaluation, Experiments, Datasets, Annotation queue, Prompt engineering, Prompt library, Playground, Optimization (selected), and Production. A message bar at the top right says "OpikAssist Beta".

The main area displays a "Trace - 8 spans" section. It lists spans under categories: agent, tools, and agent again. The first "agent" span is for "ChatGoogleGenerativeAI" and has a duration of 2.6s. The second "agent" span is for "should_continue" and has a duration of 0s. The "tools" span is for "execute_sql_query" and has a duration of 0.2s.

The "Output" panel shows a message from OpikAssist: "Alright Paddy, let's take a look at your recent progress! Based on your last 20 practice sessions, I've noticed some interesting trends: Pitch Accuracy, Timing Stability, and Scale Conformity. You've been exploring a good variety of scales including E Major, A Minor, G Major, D Minor, C# Minor, C Major, and E Minor. It's great to see you challenging yourself with different keys!"

The "OpikAssist" panel on the right provides a summary: "After a thorough scan of the trace and span details, here's what looks suspicious or potentially problematic: 1. Excessive LLM Token Usage and Long Duration: The ChatGoogleGenerativeAI span (span: 019bf07c-ca9b-a62-a841-7a486ab0d7339) consumed 4,877 tokens in one response and took almost 8 seconds (7,873 ms), making it by far the most expensive and time-consuming span in the trace. Its prompt was very lengthy, most of which appears to be context, guidelines, and the result of a prior SQL data fetch. Most of it is repeated from an earlier LLM span, indicating ineffective context management and opportunity for prompt reuse."

OpikAssist analyzing trace for token usage optimization

9. Project-Specific Configurations

Configured custom feedback definitions and AI providers for comprehensive evaluation.

Feedback Definitions:

- Custom fields for manual LLM output rating
- Human-in-the-loop feedback on traces
- Categorical ratings: "AI Coach Conversation Rating" and "User Feedback"

AI Providers:

- Perplexity's Sonar Pro for automated evaluations
- OpenRouter models for diverse evaluation perspectives
- Enhanced evaluation capabilities beyond default Opik models

The screenshot shows the "Feedback definitions" page. At the top, there are tabs for "Feedback definitions" (selected), AI Providers, Workspace preferences, and Members.

A message bar says: "Create custom fields to manually rate LLM outputs. Use them to collect structured feedback and track quality over time." Below is a search bar and a "Create new feedback definition" button.

The main table lists three feedback definitions:

	Feedback score	Type	Values	
<input type="checkbox"/>	AI Coach Conversation Rating	Categorical	Average, Bad, Excellent, Good, Very Good	...
<input type="checkbox"/>	User Feedback	Categorical	👍, 👎	...

Custom feedback definitions for manual trace ratings

Feedback definitions	AI Providers	Workspace preferences	Members
ⓘ Connect AI providers to test prompts, preview model responses, and score traces using online evaluation rules in the Playground. ×			
<input type="text" value="Q Search by name"/>			Add configuration
≡ Name	≡ URL	≡ Created	≡ Provider
Perplexity	https://api.perplexity.ai	01/28/26 08:36 PM	▼ Perplexity ...
OPENROUTER_API_KEY	-	01/28/26 08:33 PM	◀ OpenRouter ...
OPIK_FREE_MODEL_API_KEY	-		○ Opik Read-only provider

Custom AI providers configured for automated evaluations

10. Online Evaluation

Configured **11 online evaluation rules** to automatically score production traces using LLM-as-a-Judge metrics.

Purpose:

- Real-time quality monitoring of AI responses in production
- Automatic evaluation without manual review
- Early detection of performance degradation or quality issues

Rules Overview:

Hub Coach (7 rules):

1. hub_answer_correctness - Validates factual accuracy
2. hub_data_groundedness - Ensures grounding in database context
3. hub_context_usage_quality - Checks effective use of retrieved data
4. hub_actionability - Measures actionable guidance
5. hub_response_clarity - Evaluates readability
6. hub_conversational_coherence - Tracks conversation flow (thread-level)
7. hub_user_frustration_score - Detects user frustration (thread-level)

Studio AI Mode (4 rules):

8. studio_practice_recommendation_alignment - Validates goal alignment
9. studio_immediate_actionability - Ensures executable recommendations
10. studio_live_coach_feedback_quality - Measures real-time coaching quality
11. studio_live_feedback_effectiveness - Tracks session improvement (thread-level)

[View complete rule prompts and variable mappings →](#)

The screenshot shows a web-based interface for managing online evaluation rules. At the top, there's a header bar with a logo, a search bar containing 'padmanabhan-r-7119', and a link to 'Online evaluation'. To the right are buttons for 'Upgrade' and a user profile. Below the header is a sidebar with various icons. The main area is titled 'Online evaluation' and contains a sub-header: 'Automatically score your production traces by defining LLM-as-a-Judge or code metrics. [Read more](#)'. A search bar with placeholder 'Search by ID' and a 'Create new rule' button are at the top of the list table. The table has columns: 'Name', 'Projects', 'Scope', 'Status', and 'Logs'. Each row represents a different rule, such as 'hub_answer_correctness' or 'studio_live_feedback_effectiveness', with details like project names (FretCoach, FretCoach-Hub), scope (Trace or Thread), status (Enabled), and a 'Show logs' button.

Online evaluation rules dashboard (1-10 of 11)

This screenshot shows the continuation of the online evaluation rules dashboard, specifically rules 11 through 11. The interface is identical to the previous screenshot, featuring a sidebar, a header with a logo and search bar, and a main table for listing rules. The table shows one rule: 'hub_user_frustration_score' with a status of 'Enabled'. The footer indicates 'Showing 11-11 of 11'.

Online evaluation rules dashboard (11 of 11)

11. Production Dashboard

A real-time dashboard monitoring key AI quality metrics across FretCoach's Studio and Hub applications.

[View Live Dashboard](#)

Dashboard Structure:

The dashboard displays 7 core metrics organized by application:

FretCoach - Studio and Portable (Core Functionality)

Evaluates practice recommendations and real-time coaching effectiveness.

Metric	Range	What the Score Means
Live Coach Feedback Quality	1 – 4	1 = Bad, 2 = Good, 3 = Very Good, 4 = Excellent
Practice Recommendation Alignment	0.0 – 1.0	1.0 = Aligned with the player's weaknesses; 0.0 = No alignment

⚡ Practice Recommendation - Immediate Actionability	0.0 – 1.0	1.0 = Recommendation fully actionable; 0.0 = Not actionable
--	------------------	---

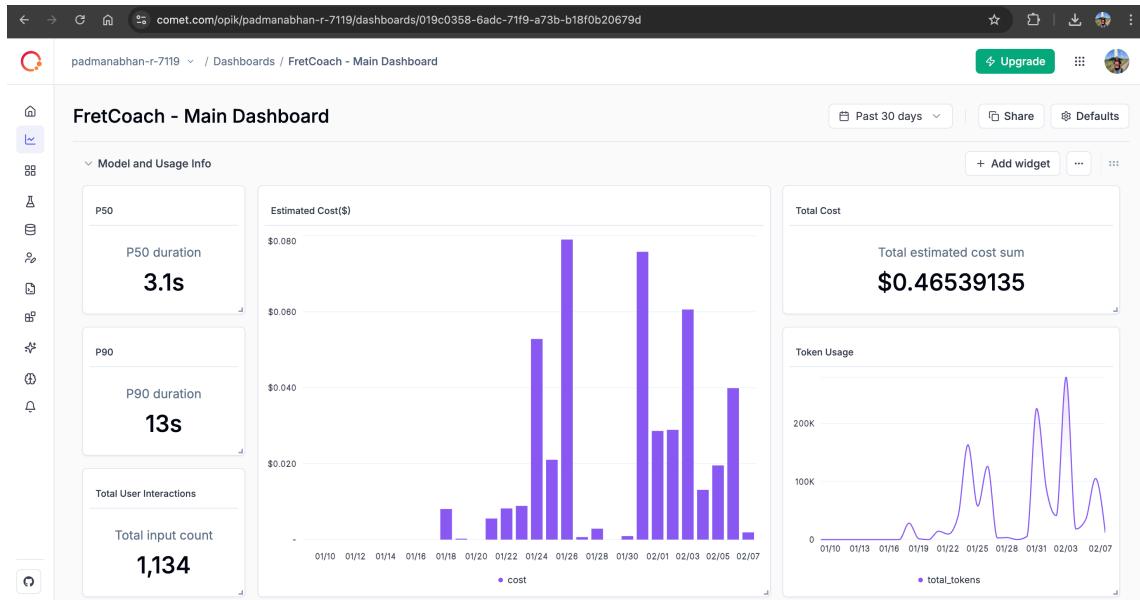
🧠 FretCoach Hub (Web)

Measures how well the Hub understands, answers, and guides users.

Metric	Range	What the Score Means
🌟 Response Clarity	0.0 – 1.0	1.0 = Clear and structured response; 0.0 = Poor response
🔗 Context Usage Quality	0.0 – 1.0	1.0 = Context effectively used; 0.0 = Poor usage of context
🎯 Actionability	0.0 – 1.0	1.0 = Clear, executable next steps; 0.0 = Vague or non-actionable
📊 Data Groundedness	0.0 – 1.0	1.0 = Supported by user's practice data; 0.0 = Weak grounding

Features:

- Real-time metric averages calculated from production traces
- Automatic updates as new LLM calls are evaluated
- Clear visibility into AI quality across different use cases
- Easy identification of performance degradation



Dashboard overview - Model and usage statistics

The screenshot shows a detailed view of the FretCoach - Main Dashboard. At the top, there's a navigation bar with a user icon, a dropdown menu, and links for 'Dashboards' and 'FretCoach - Main Dashboard'. On the right side of the header are buttons for 'Upgrade', 'Share', 'Defaults', and a three-dot menu. Below the header, there's a search bar with a dropdown for 'Past 30 days', a 'Share' button, and a 'Defaults' button. A 'FretCoach - AI Evaluation Metrics Info' section is expanded, titled 'FretCoach - AI Evaluation Metrics'. It contains two tables: one for 'FretCoach - Studio and Portable (Core Functionality)' and another for 'FretCoach Hub (Web)'. Both tables provide metric details, ranges, and score meanings.

AI Evaluation Metrics documentation embedded in dashboard

This screenshot shows the live production dashboard for FretCoach. The layout is similar to the main dashboard, with a sidebar on the left and various sections on the right. The 'FretCoach - Main Dashboard' section is visible at the top. Below it, the 'FretCoach - Studio and Portable (Core Functionality)' section is expanded, showing four key metrics with their average values: 'Average Live Coach Feedback Quality' (3.432), 'Average Practice Recommendation Alignment' (0.884), 'Average Practice Recommendation - Immediate Actionability' (0.947), and 'Average Hub Response Clarity' (0.972). The 'FretCoach Hub (Web)' section is also expanded, showing 'Average Hub Context Usage Quality' (0.97), 'Average Hub Actionability' (0.994), and 'Average Hub Data Groundedness' (0.834). A 'Add section' button is located at the bottom center of the dashboard area.

Live production dashboard with real-time AI quality metrics

12. Alerts & Notifications

Configured Slack alerts to proactively monitor AI quality and system health in production.

Setup:

- Created a dedicated Slack channel: #opik-alerts
- Integrated Opik with Slack using webhook configuration

- Configured alerts for critical metrics and system errors

Alert Types:

1. Trace Errors Threshold

- Trigger:** When trace error count exceeds 10 in the last 30 minutes
- Purpose:** Detect system failures or integration issues

2. Feedback Score Thresholds

- Trigger:** When average metric scores fall below 0.6 in the last 30 minutes
- Monitored Metrics:**
 - Hub Response Clarity < 0.6
 - Hub Data Groundedness < 0.6
 - Hub Context Usage Quality < 0.6
 - Hub Answer Correctness < 0.6
 - Hub Actionability < 0.6
- Purpose:** Early detection of AI quality degradation

3. Latency Alerts

- Trigger:** When average latency exceeds 3 seconds in the last 30 minutes
- Purpose:** Monitor response time performance and identify slowdowns

Benefits:

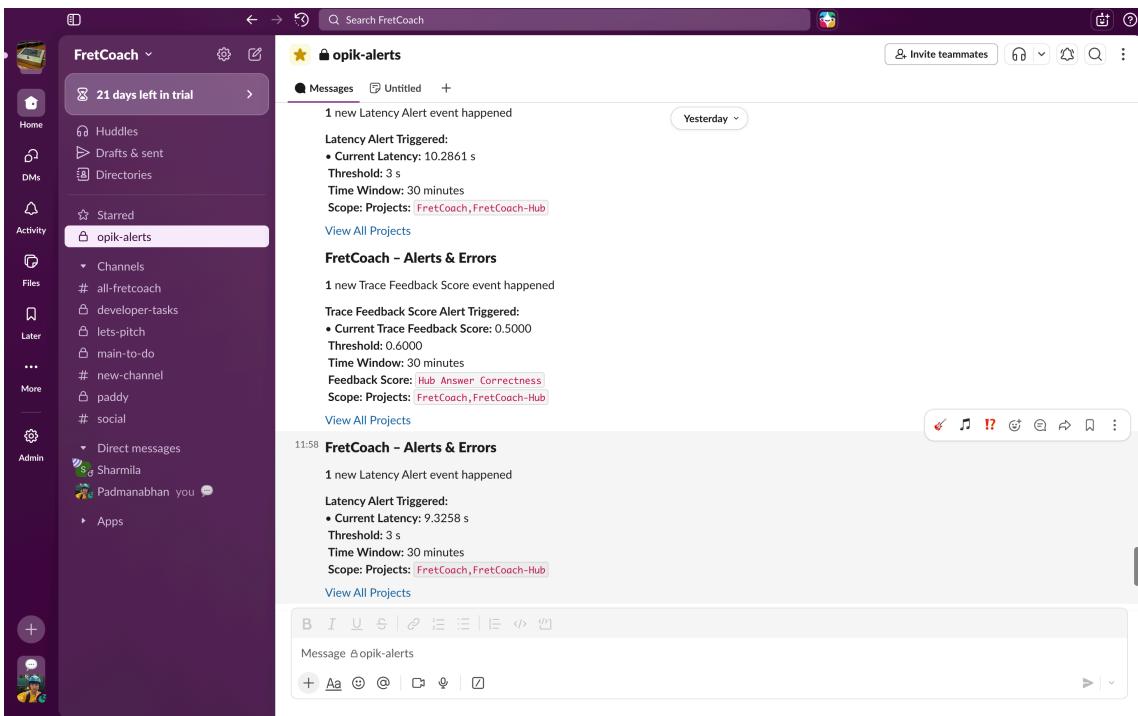
- Proactive issue detection before users report problems
- Real-time visibility into production AI quality
- Team-wide awareness through Slack notifications
- Quick response to quality degradation or system errors

```

1 v {
  "blocks": [
    {
      "type": "header",
      "text": {
        "type": "plain_text",
        "text": "FretCoach - Alerts & Errors"
      }
    },
    {
      "type": "section",
      "text": {
        "type": "mrkdwn",
        "text": "*Trace Errors Alert Triggered*\n*Current Trace Errors*: 15\n*Threshold*: 10\n*Time Window*: 1 hour\n*Scope*: *Projects*: *Demo Project, Default Project*\n*Project*: *\n*Project URL*: http://localhost:5173/demo_workspace_name/projects\n*View All Projects*"
      }
    }
  ]
}

```

Alert configuration in Opik dashboard



Real-time alerts delivered to Slack #opik-alerts channel

13. Key Insights Gained from Production Observability

Opik traces surfaced actionable improvements that directly improved FretCoach's quality and performance:

Insight	Discovery	Action Taken	Result
Prompt verbosity	Live coach prompts were verbose, causing slow responses	Tightened prompt to "1-2 sentences, max 30 words" constraint	Significantly faster responses, more focused feedback
TTS latency spike	TTS taking longer than expected on some calls	Implemented singleton audio player instance to prevent concurrent playback	Consistent TTS latency, no audio overlap
Prompt optimization	Live feedback prompt quality measured via <code>llm_judge_metric</code>	Used Optimization Studio (HRPO) to refine the prompt	32% increase in live coaching response quality
Fallback model visibility	~15% of Hub Chat requests hit Gemini rate limits	Confirmed MiniMax fallback working seamlessly, kept hybrid approach	Zero user-facing errors on rate limit
Token cost patterns	Different features had very different token footprints	Targeted gpt-40-mini for cost-sensitive real-time features	Optimized cost-performance ratio per feature

Ongoing: All 11 online evaluation rules continue to run in production, monitoring AI quality across both Studio and Hub with automatic alerts when scores degrade.