

# 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 natural language web dashboard agent. 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 and organization.

#### 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 various project and tool options. The 'Projects' section is currently active. The main content area shows a 'Trace - 18 spans' section with a tree view of the trace structure. Spans include 'LangGraph', 'agent', 'should\_continue', 'tools', and 'execute\_sql\_query'. To the right, there's a detailed view of the trace structure with nodes for 'start', 'agent', 'end', and 'tools'. Below this, a 'Feedback scores' table is shown with columns for 'Key', 'Score', and 'Reason'. The table includes rows for Hub Response Clarity, Hub Answer Correctness, Hub Actionability, Hub Data Groundedness, and Hub Context Usage Quality, all rated as 1.

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

Trace - 2 spans

**RunnableSequence**

- 01/31/26 12:41 PM 3.1s # 2084 <\$0.001 ↗ 2
- ai-mode fretcoach-core fretcoach-studio gpt-4o-mini practice-recommendation

**Details** Feedback scores

Input

```
Pretty: You are an AI guitar coach. Based on the practice history below, recommend a practice session.
```

PRACTICE HISTORY:

- Total sessions: 8
- Average pitch accuracy: 71.6%
- Average scale conformity: 21.1%
- Average timing stability: 56.5%
- Weakest area: scale

RECENTLY PRACTICED SCALES:

```
[{"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

Trace - 1 spans

**ChatOpenAI**

- 01/30/26 10:49 PM 2s # 389 <\$0.001 ↗ 1 ↗ 5
- fretcoach-core fretcoach-studio gpt-4o-mini live-feedback manual-mode

**Details** Feedback scores

Input

```
Pretty: Enabled metrics: Pitch Accuracy, Timing Stability  
Pitch 95%, Timing 21%  
Strongest: Pitch Accuracy (95%)  
Weakest: Timing Stability (21%)
```

Give 1-2 sentences (max 30 words) - what's good, what's weak, specific actionable fix:

Output

```
Pretty: Your pitch accuracy is outstanding at 95%, but timing stability is lacking at 21%. Slow down and count to improve your rhythm consistency.
```

Metadata

```
YAML: providers:
```

### 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-{user_id}` - 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 threads. Each thread entry includes a unique ID, a timestamped message, and a JSON snippet of the message content. The threads are grouped by their type, such as 'hub-aicoach-chat'. The interface has a search bar, filters, and a timeline selector.

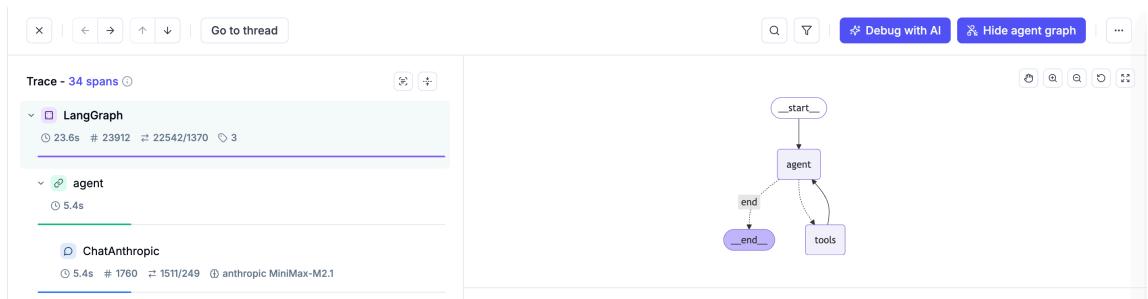
ID	Message	Content Snippet
hub-aicoach-chat-1770250552190	How am I doing compared to average?	{"messages": [{"content": "Hi! I'm your AI practice co..."}]
hub-aicoach-chat-1770223524432	How am I doing compared to average? And what day is ...	{"messages": [{"content": "Hi! I'm your AI practice co..."}]
hub-aicoach-chat-1770217604585	How am I doing compared to average?	{"messages": [{"content": "Hi! I'm your AI practice co..."}]
hub-aicoach-chat-1770215363136	What should I practice today?	```json { "focus_area": "Scale Conformity", "current...`
hub-aicoach-chat-1770215206179	What should I practice today?	{"messages": [{"content": "Hi! I'm your AI practice co..."}]
e868ca6e-c204-45e8-83c0-0be930248505-live-aicoach-feedback	Enabled metrics: Pitch Accuracy, Scale Conformity, T...	Timing is spot on at 100%, but scale conformity is...
6db7f5e0-1bdd-484d-9b1b-51059b3f8e90-live-aicoach-feedback	Enabled metrics: Pitch Accuracy, Scale Conformity, T...	Timing is excellent at 97%, but scale conformity is...
0a7d4419-a110-4ff7-a8b6-ae56363283d7-live-aicoach-feedback	Enabled metrics: Pitch Accuracy, Scale Conformity, T...	Timing is excellent at 98%, but scale conformity is...
hub-aicoach-chat-1770035493514	What should I practice today?	{"messages": [{"content": "Hi! I'm your AI practice co..."}]
hub-aicoach-chat-1770035435207	What should I practice today?	{"messages": [{"content": "Hi! I'm your AI practice co..."}]
hub-aicoach-chat-1770035366500	What should I practice today?	{"messages": [{"content": "Hi! I'm your AI practice co..."}]
hub-aicoach-chat-1770035155059	Show me my progress	{"messages": [{"content": "Hi! I'm your AI practice co..."}]}

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 a web-based annotation interface for a thread quality check. At the top, there's a navigation bar with links to 'Study', 'Blogs', 'Github', 'Great Learning', and 'Competitor Proj...'. Below that is a header for 'AI Coach Chat Thread Quality Checks' with a date ('01/28/26 09:12 PM'), a dropdown for 'Threads', a search bar ('FretCoach-Hub'), and a progress indicator ('28/28 (100%)'). There are also buttons for 'Upgrade', 'Share', 'Edit', 'Export queue', and 'Annotate'.

The main area is titled 'Queue items' and contains a table with the following columns: ID, First message, Last message, and Comments. The table lists six threads, each with a checkbox, a message from the user, a response from the AI, and a red circular icon with a minus sign indicating a problem. The comments column provides detailed feedback for each thread.

ID	First message	Last message	Comments
thread-1769613376415	What should I practice today? I feel like doing some G# 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.

### Datasets:

- Curated test cases from real user sessions
- Used across experiment runs for consistent testing

### Prompts:

- Version-controlled coaching prompt templates
- Used in playground for rapid iteration

Datasets created for experiment runs

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	-	-		

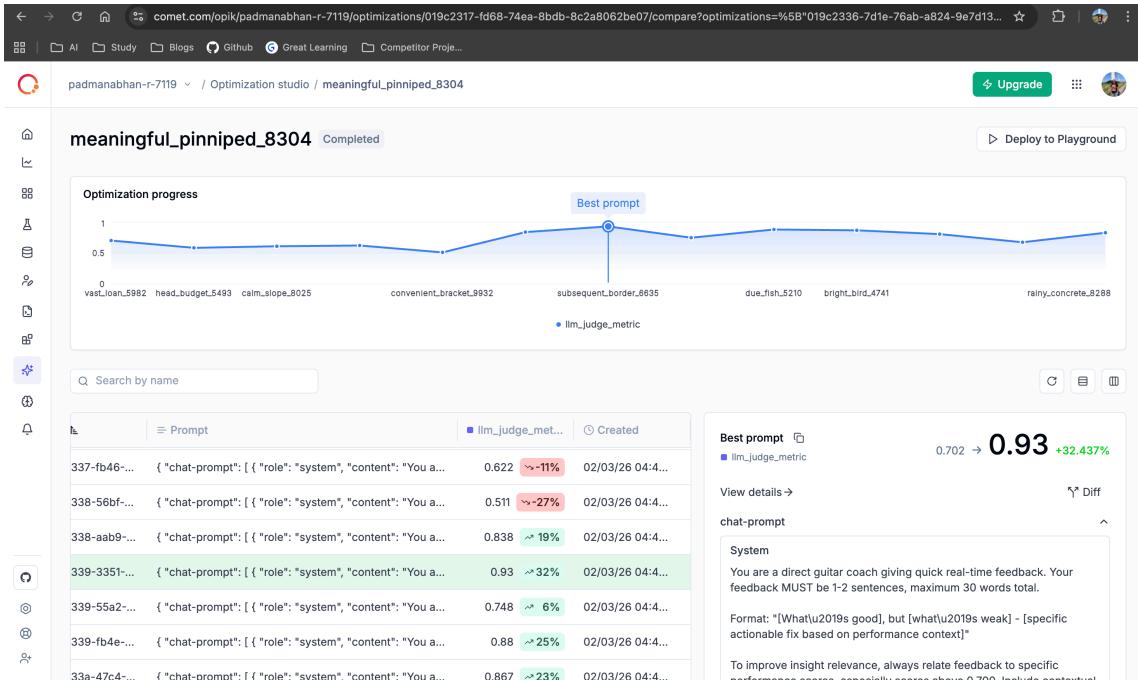
### Experiments with custom metric

## 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 is a sidebar with navigation links like Home, Dashboards, Observability, Projects (which is selected), Evaluation, Experiments, Datasets, Annotation queue, Prompt engineering, Prompt library, Playground, Optimization, and Production. The main area has tabs for 'Trace - 8 spans' and 'Output'. The 'Output' tab displays a message from 'OpikAssist Beta' suggesting improvements based on recent practice sessions. It highlights excessive LLM token usage and long duration, specifically mentioning a span related to ChatGoogleGenerativeAI. A text input field at the bottom allows users to type messages to the AI.

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 are tabs for 'Feedback definitions' (selected), 'AI Providers', 'Workspace preferences', and 'Members'. Below is a search bar and a button to 'Create new feedback definition'. A note says 'Create custom fields to manually rate LLM outputs. Use them to collect structured feedback and track quality over time.' The main area lists three feedback definitions: 'Feedback score' (Type: Categorical, Values: Average, Bad, Excellent, Good, Very Good), 'AI Coach Conversation Rating' (Type: Categorical, Values: Average, Bad, Excellent, Good, Very Good), and 'User Feedback' (Type: Categorical, Values: thumbs up, thumbs down).

Custom feedback definitions for manual trace ratings

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

*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, and a 'Upgrade' button. 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 labeled 'Search by ID' is present. To the right of the search bar are buttons for 'Create new rule' and other actions. The main content is a table with columns: 'Name', 'Projects', 'Scope', 'Status', and 'Logs'. The table lists 11 rules, all of which are enabled. Each row includes a checkbox, a name like 'hub\_answer\_correctness', project names like 'FretCoach, FretCoach-Hub', scope type (Trace or Thread), status (Enabled), and a 'Show logs' button.

Online evaluation rules dashboard (1-10 of 11)

This screenshot shows the same dashboard as the previous one, but it is labeled 'Showing 11-11 of 11', indicating that all 11 rules have been loaded. The table structure is identical, showing 11 rows of rules, all enabled, with columns for Name, Projects, Scope, Status, and Logs.

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
<b>Live Coach Feedback Quality</b>	<b>1 – 4</b>	1 = Bad, 2 = Good, 3 = Very Good, 4 = Excellent
<b>Practice Recommendation Alignment</b>	<b>0.0 – 1.0</b>	1.0 = Aligned with the player's weaknesses; 0.0 = No alignment

<b>⚡ Practice Recommendation - Immediate Actionability</b>	<b>0.0 – 1.0</b>	1.0 = Recommendation fully actionable; 0.0 = Not actionable
--	------------------	---

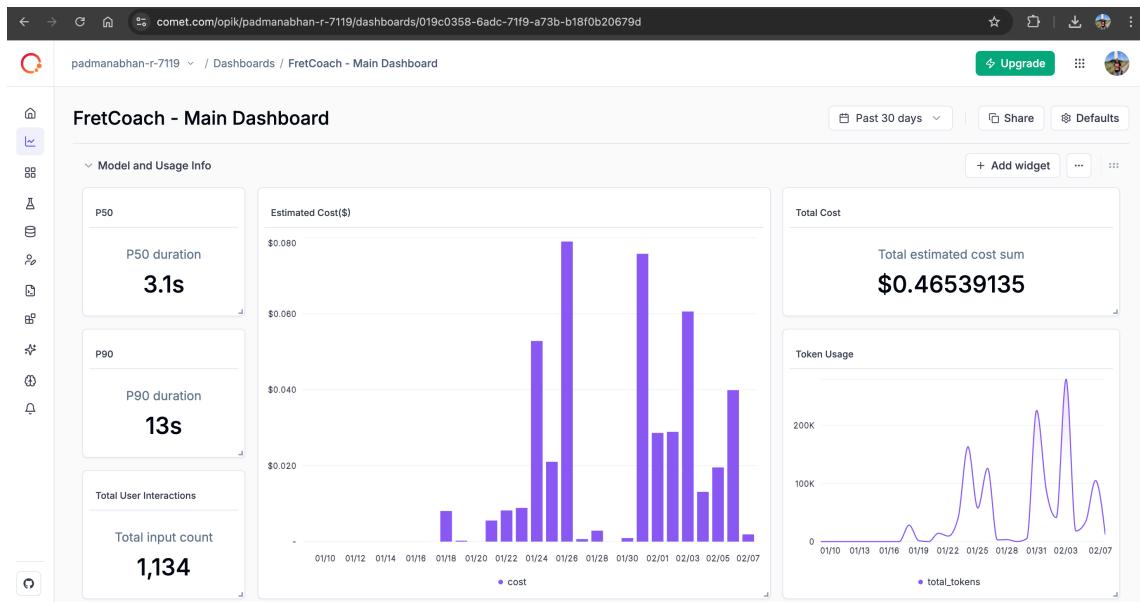
### 🧠 FretCoach Hub (Web)

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', and 'Defaults'. 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 three cards with numerical scores: 'Average Live Coach Feedback Quality' (3.432), 'Average Practice Recommendation Alignment' (0.884), and 'Average Practice Recommendation - Immediate Actionability' (0.947). The 'FretCoach Hub (Web)' section is also expanded, showing four cards with numerical scores: 'Average Hub Response Clarity' (0.972), 'Average Hub Context Usage Quality' (0.97), 'Average Hub Actionability' (0.994), and 'Average Hub Data Groundedness' (0.834). Each card includes a brief description of the metric and its scale.

*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

The screenshot shows the Opik dashboard interface for managing alerts. On the left, there are several alert configurations listed:

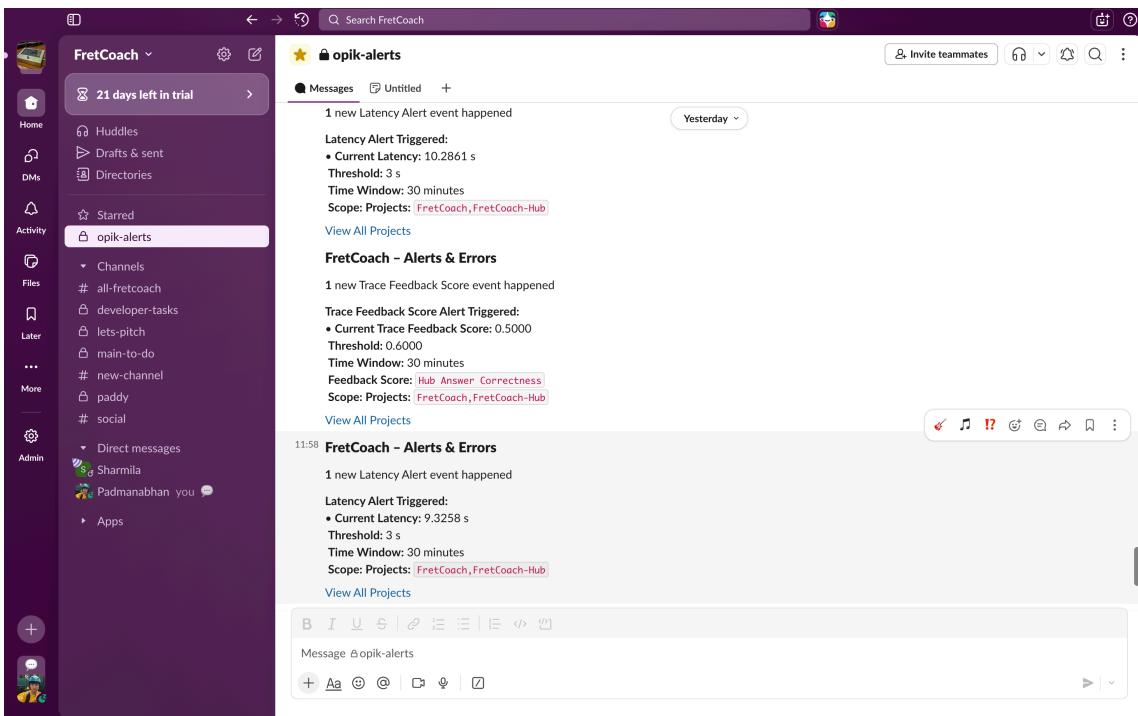
- Trace errors threshold:** Triggered when the number of trace errors exceeds the specified threshold in selected projects. Set to trigger when the count exceeds 10 in the last 30 minutes.
- Trace feedback score threshold:** Triggered when the average feedback score for traces exceeds the specified threshold in selected projects. Conditions include Hub Response Clarity < 0.6, Hub Data Groundedness < 0.6, Hub Context Usage Quality < 0.6, Hub Answer Correctness < 0.6, and Hub Actionability < 0.6, all within the last 30 minutes.
- Thread feedback score threshold:** Triggered when the average feedback score for threads exceeds the specified threshold in selected projects. Set to trigger when the average is less than 0.6 in the last 30 minutes.

On the right, a detailed view of the "Trace errors threshold" alert configuration is shown. It includes a "Test alert configuration" section with a "Test connection" button and a "Go to docs" link. The "Payload" section displays the JSON configuration for the alert:

```

1 v {
2 v   "blocks": [
3 v     {
4 v       "type": "header",
5 v       "text": {
6 v         "type": "plain_text",
7 v         "text": "FretCoach - Alerts & Errors"
8 v       }
9 v     },
10 v     {
11 v       "type": "section",
12 v       "text": {
13 v         "type": "mrkdwn",
14 v         "text": "*Text*: **1* new Trace Error Alert event happened"
15 v       }
16 v     },
17 v     {
18 v       "type": "section",
19 v       "text": {
20 v         "type": "mrkdwn",
21 v         "text": "*Trace Errors Alert Triggered*:\\n\\n*Current Trace Errors*: 15\\n *Threshold*: 10\\n *Time Window*: 1 hour\\n *Scope*: *Projects*: 'Demo Project,Default Project'\\n\\nhttp://localhost:5173/demo_workspace_name/projects\\n\\nAll Projects"
22 v       }
23 v     }
24 v   }
25 v }
```

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
<b>Prompt verbosity</b>	Live coach prompts were verbose, causing slow responses	Tightened prompt to "1-2 sentences, max 30 words" constraint	Significantly faster responses, more focused feedback
<b>TTS latency spike</b>	TTS taking longer than expected on some calls	Implemented singleton audio player instance to prevent concurrent playback	Consistent TTS latency, no audio overlap
<b>Prompt optimization</b>	Live feedback prompt quality measured via <code>llm_judge_metric</code>	Used Optimization Studio (HRPO) to refine the prompt	<b>32% increase</b> in live coaching response quality
<b>Fallback model visibility</b>	~15% of Hub Chat requests hit Gemini rate limits	Confirmed MiniMax fallback working seamlessly, kept hybrid approach	Zero user-facing errors on rate limit
<b>Token cost patterns</b>	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.