

# Opik Integration in FretCoach

Workspace: padmanabhan-r-7119 Projects: FretCoach | FretCoach-Hub

## 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 web interface. On the left, a sidebar menu includes Home, Dashboards, Observability, Projects (selected), Evaluation, Experiments, Datasets, Annotation queue, Prompt engineering, Prompt library, Playground, Optimization, Production, Online evaluation, Alerts, Star (17.5K), Configuration, Support hub, and Invite a teammate. The main area displays a trace analysis for 18 spans. A detailed breakdown of one span shows it took 14.6s, involved 15146 API calls, cost \$0.009, and was labeled with tags: ai-coach-chat, fretcoach-hub, from-hub-dashboard, gemini-2.5-flash, and practice-plan. Below this, there's a LangGraph visualization showing a flow from start to agent, then to tools, and back to end. A feedback scores section provides a summary of the AI's performance across five categories: Hub Response Clarity, Hub Answer Correctness, Hub Actionability, Hub Data Groundedness, and Hub Context Usage Quality, each with a score of 1 and a brief reason. 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

Trace - 2 spans

RunnableSequence

3.1s # 2084 1997/87 <\$0.001 2 5

ChatOpenAI

3.1s # 2084 1997/87 <\$0.001 openai gpt-4o-mini-2024-07-18

RunnableLambda

0.00ts

RunnableSequence

3.1s # 2084 1997/87 <\$0.001 2 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

The screenshot shows the Comet AI interface for monitoring machine learning models. On the left, a sidebar lists various monitoring categories like Model, Observability, Evaluation, Promotions, and Optimizations. The main dashboard displays a trace analysis for a 'ChatOpenAI' model. A summary card for 'ChatOpenAI' shows metrics: 2s duration, 389 samples, 361/28 errors, <\$0.001 cost, 1 warning, and 5 feedback scores. Below this, a detailed view for a specific trace entry also shows these metrics. The detailed view includes tabs for 'Details' and 'Feedback scores'. Under 'Details', there's an 'Input' section with a 'Pretty' dropdown and a 'Feedback scores' section listing enabled metrics: Pitch Accuracy, Timing Stability, Pitch 95%, Timing 21%, Strongest: Pitch Accuracy (95%), and Weakest: Timing Stability (21%). It also prompts for a 1-2 sentence actionable fix. The 'Output' section shows a message about pitch accuracy and timing stability. The 'Metadata' section includes a 'YAML' dropdown and a 'providers:' field.

### *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

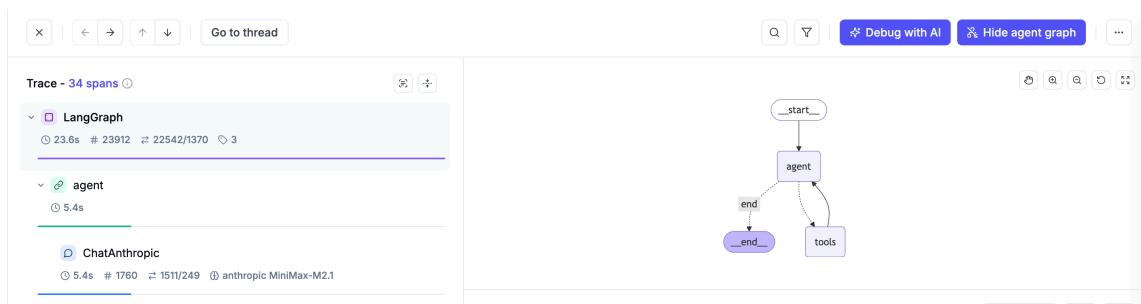
The screenshot shows a web-based application interface for managing conversation threads. At the top, there's a navigation bar with links like 'Study', 'Blogs', 'Github', and 'Competitor Proj...'. Below the navigation is a search bar with placeholder text 'Search by ID' and filters for 'First message' and 'Last message'. The main content area lists several thread IDs, each with a timestamp, message content, and JSON representation of the messages. The threads are categorized by their purpose, such as 'hub-aicoach-chat' for AI practice and 'aicoach-feedback' for live feedback. The interface is clean with a light blue header and white background.

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 for '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 five threads, each with a detailed message history and a red circular icon with a minus sign indicating a problem. The messages show AI responses like "Hi! I'm your AI practice coach. I can help you analyze your performance, suggest exercises, and answer questions about your practice". The comments column includes entries such as "Very bad, no practice plan actually displayed.", "Add guard rails, as it is responding to random texts", and "Not many details".

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	-	-		

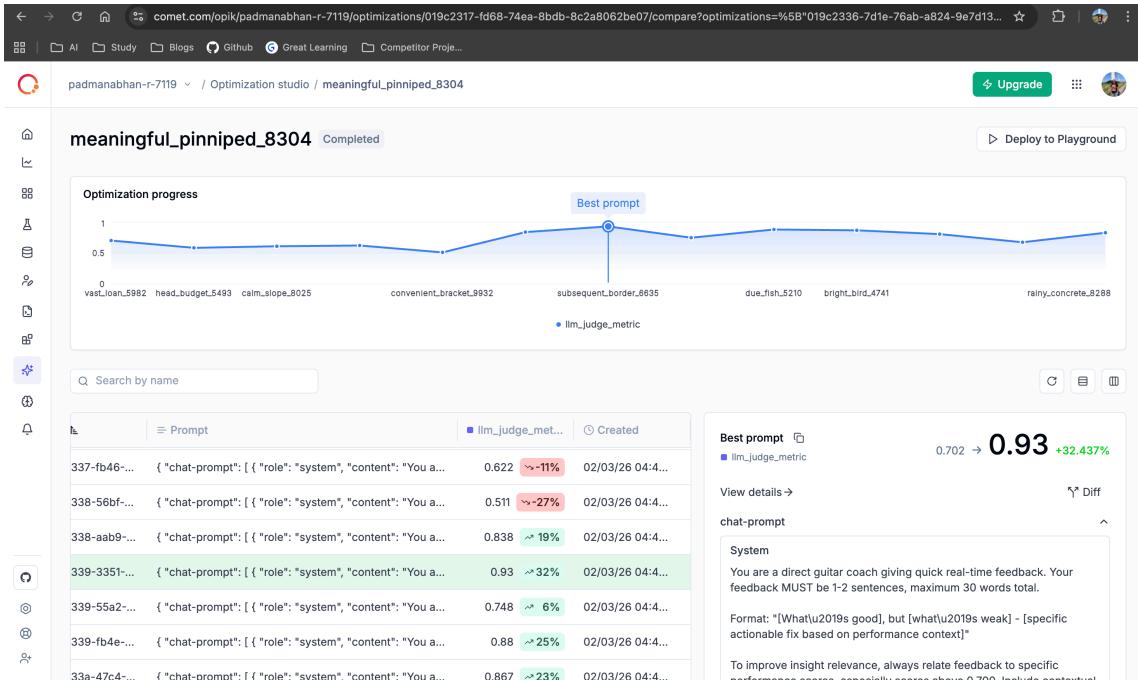
### Experiments with custom metric

## 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, 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 "Debug with AI".

The main area displays a "Trace - 8 spans" section. It lists spans under categories: agent, tools, and agent again. Key spans include "ChatGoogleGenerativeAI" (2.6s), "should\_continue" (0s), "execute\_sql\_query" (0.2s), and another "ChatGoogleGenerativeAI" (7.9s). An "Output" panel shows a message from "Pretty" suggesting improvements in pitch accuracy, timing stability, and scale conformity.

To the right, the "OpikAssist Beta" panel provides a summary of findings: "Excessive LLM Token Usage and Long Duration" (specifically for the ChatGoogleGenerativeAI span) and "Its prompt was very lengthy, most of which appears to be context, guidelines, and the result of a prior SQL data fetch". It also notes that the span consumed 4,877 tokens and took 8 seconds.

*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, tabs include "Feedback definitions" (selected), AI Providers, Workspace preferences, and Members. A note 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.

	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	<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, each with a checkbox, a name, a project list, a scope type (Trace or Thread), an enabled status, a 'Show logs' link, and a three-dot menu. At the bottom of the table, it says 'Showing 1-10 of 11'.

*Online evaluation rules dashboard (1-10 of 11)*

This screenshot shows the final page of the online evaluation rules dashboard, displaying rules 11 through 11. The interface is identical to the previous screenshot, with a sidebar, a header, and a main table. The table has one visible row for 'hub\_user\_frustration\_score' with the same columns and details as the previous rows. At the bottom, it says '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
<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
--	------------------	---

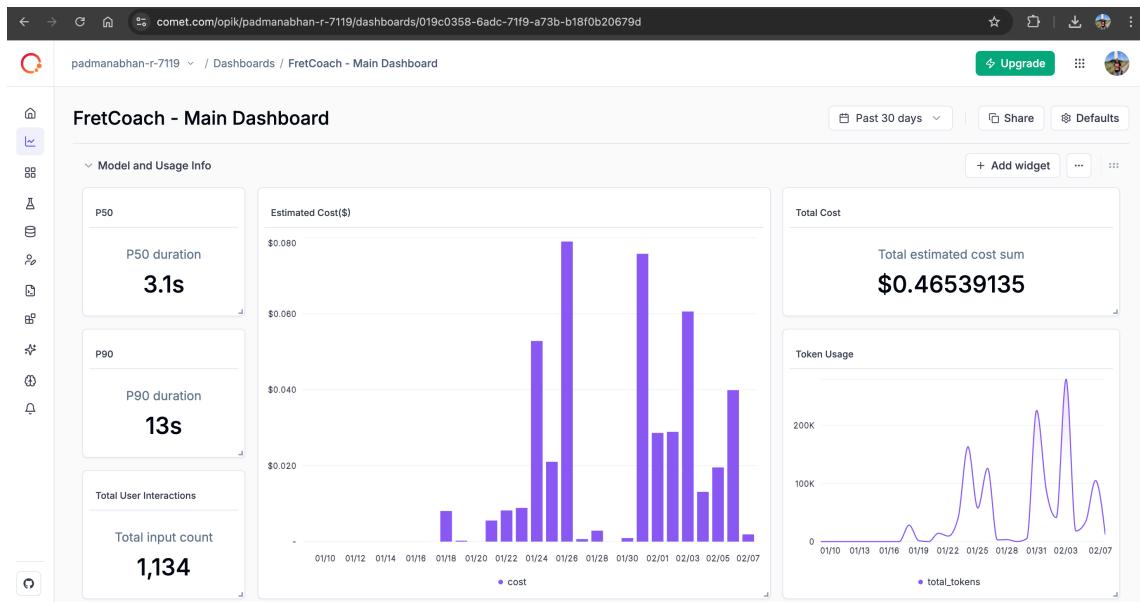
### 🧠 FretCoach Hub (Web)

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 buttons for 'Upgrade', 'Share', and 'Defaults'. Below the header, a section titled 'FretCoach - AI Evaluation Metrics Info' contains two main sections: 'FretCoach - Studio and Portable (Core Functionality)' and 'FretCoach Hub (Web)'. Each section includes a brief description, a table of metrics with ranges and meanings, and a 'What the Score Means' table.

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

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

*AI Evaluation Metrics documentation embedded in dashboard*

This screenshot shows the live production dashboard for FretCoach. It features a similar layout to the main dashboard but with more widgets. The top section displays three key metrics: Average Live Coach Feedback Quality (3.432), Average Practice Recommendation Alignment (0.884), and Average Immediate Actionability (0.947). Below these, there are four more sections: Response Clarity (0.972), Context Usage Quality (0.97), Average Hub Actionability (0.994), and Average Hub Data Groundedness (0.834). Each section 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:

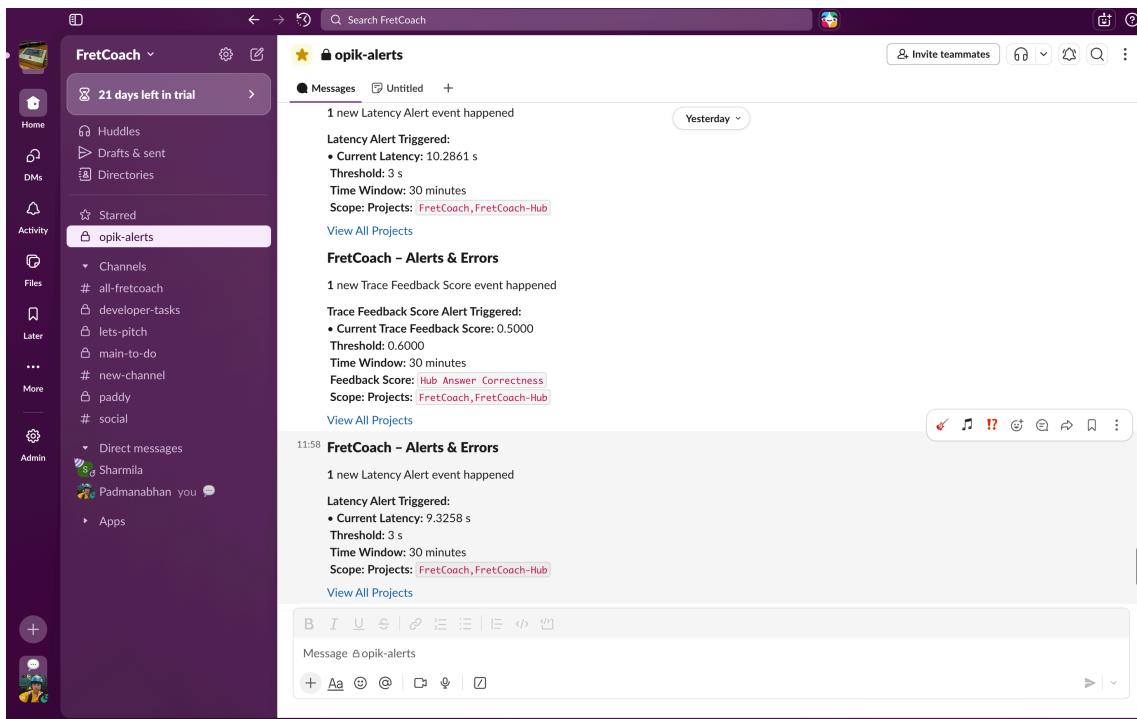
- Proactive issue detection before users report problems
- Real-time visibility into production AI quality
- Team-wide awareness through Slack notifications
- Quick response to quality degradation or system errors

```

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