# 📄 FSD_09 – Admin Review Dashboard + Feedback Router

---

## 🧠 Purpose:

This module provides **Audira admins and partner consultants** with a secure, centralized interface for:

- Reviewing agent readiness
- Overriding failed validator outputs
- Managing flagged memory or conflict cases
- Routing feedback and user corrections into the retraining pipeline

It acts as the **human-in-the-loop oversight layer** and feedback collection system for quality control and continuous learning.

---

## 🧩 FSD_09 – Section Breakdown

| Section | Description |
|---|---|
| **1. Scope** | What the admin dashboard and feedback system are responsible for |
| **2. Input Sources** | What data feeds into the dashboard and router |
| **3. Dashboard Views & Controls** | What admins can see, do, override, or flag |
| **4. Feedback Collection Points** | Where feedback is gathered (users, partners, QA) |
| **5. Feedback Processing Logic** | How input is interpreted, stored, and escalated |
| **6. Escalation & Triage Logic** | How issues are resolved or routed internally |
| **7. Retraining Hooks** | How confirmed corrections are fed into agent learning or retraining |
| **8. Future Enhancements** | Audit trail visibility, auto-labeling, QA scoring, and agent benchmarking |

## 🔷 Section 1: Scope

---

### 🎯 Purpose:

This module provides the **control panel** for internal reviewers, consultants, or authorized partners to:

- Inspect the full onboarding results of any agent
- View readiness status and validator output
- Manually override failed launch checks with justification
- Resolve memory conflicts or contradictions
- Route flagged issues to the correct team (content, QA, retraining)
- Collect structured feedback from users, testers, or field partners

It is **not a passive viewer** — it is an active governance interface in the Audira platform lifecycle.

---

### ✅ Responsibilities:

| Function | Description |
|---|---|
| **Agent Readiness Review** | View detailed validator reports (from FSD_05), tag coverage, segment usage |
| **Override Gatekeeper** | Let authorized users force agent launch with justification |
| **Memory Conflict Manager** | Inspect, resolve, or flag contradictory agent facts (from FSD_07) |
| **Feedback Router** | Collect and triage user or QA feedback during or after agent use |
| **Retraining Trigger Engine** | Flag agents or blocks for AI model fine-tuning based on structured signals |
| **Audit & Compliance Logging** | Maintain full trail of changes, overrides, and decision history |

---

### ❌ Not in Scope:

| Excluded Feature | Reason |
|---|---|
| **Editing onboarding answers** | That occurs in the onboarding interface |
| **Modifying uploaded documents** | Those are read-only once parsed in FSD_01 |
| **Direct agent response generation** | That is handled live in FSD_08 |
| **LLM prompting or inference logic** | Controlled by FSD_06 and FSD_08 |

---

### 🧩 Supported Reviewer Roles:

| Role | Capabilities |
|---|---|
| **Internal Admin** | Full access to override, resolve, revalidate, and mark agents for retraining |
| **Partner Consultant** | Read + resolve permissions only for assigned clients |
| **QA Team Member** | Can simulate agent behavior, submit scoring feedback |
| **Feedback Analyst** | Can review user-captured issues, but not resolve conflicts |

---

### 📎 **References**:

- 📘 *AUDIRA PRE-LAUNCH VALIDATOR SPEC* – defines readiness report and override logic
- 📘 *AUDIRA AGENT BLUEPRINT TEMPLATE* – identifies agent role, owner, partner metadata
- 📘 *AUDIRA FILE & DATA UPLOAD SCHEMA* – allows source tracing for tag conflicts
- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – provides context for simulation/test mode
- 📘 *AUDIRA PRODUCT BLUEPRINT* – defines governance, QA, and escalation process roles

---

## 🔷 Section 2: Input Sources

This section defines all the data streams that feed into the Admin Review Dashboard and Feedback Router. These inputs are pulled from earlier FSD modules and real-time interaction logs.

---

### 📦 A. Readiness Validator Output

**Source:** FSD_05
Provides overall launch readiness, score, failed areas, and auto-suggested fixes.

```
{
 "agent_id": "AGENT_00231",
 "readiness_score": 71,
 "status": "FAIL",
 "failed_areas": [
  "MISSING_CRITICAL_TAG:pricing_strategy",
  "DOCUMENT_GAPS:recency"
 ],
 "suggested_fixes": [
  "Upload a recent overview doc",
  "Answer onboarding question Q3"
 ]
}
```

---

### 📦 B. Memory Conflict Records

**Source:** FSD_07
Captures unresolved contradictions between user answers, document content, or segment analysis.

```
{
  "tag_id": "payout_policy",
  "conflict_detected": true,
  "status": "pending_resolution",
  "values": ["automated via Stripe", "manual payout by finance"]
}
```

---

## 📦 C. Agent Metadata & Intent

**Source:** Agent Onboarding Framework
Includes industry, launch type, partner owner, and module capabilities.

```
{
  "agent_id": "AGENT_00231",
  "industry": "SaaS",
  "configured_by": "partner_consultant_127",
  "capabilities": ["Q&A", "Sales"]
}
```

---

## 📦 D. Feedback Logs

**Sources:**

- Agent chat interface (user rating buttons, “Was this helpful?”)
- QA test results
- Admin simulations (FSD_10 preview hooks)

```
{
  "feedback_id": "fbk_90211",
  "source": "user_live_session",
  "linked_response": "Agent said: Vendors are paid manually.",
  "feedback_type": "incorrect_info",
```

"submitted_by": "user_342",

"timestamp": "2025-06-11T03:33:00Z"

}

---

## 📦 E. Prompt & Response Trace Logs

**Source:** FSD_08
Used for simulation replay and traceable decision flow.

{

"turn_id": "turn_8181",

"model_used": "gpt-4",

"input_prompt": "...",

"response": "You offer three pricing tiers...",

"tags_covered": ["pricing_model"],

"memory_used": true

}

---

## 🧩 Summary Input Table

| Input | Used For |
|---|---|
| **readiness_report** | Validator decision review and override logic |
| **conflict_log** | Memory inspection and contradiction resolution |
| **agent_metadata** | Display filters and partner-specific views |
| **feedback_items** | Triage for QA, retraining, or correction |
| **response_traces** | Replay and response behavior review |

---

📎 **References**:

- 📘 *AUDIRA PRE-LAUNCH VALIDATOR SPEC* – defines readiness scoring fields and logic
- 📘 *AUDIRA FILE & DATA UPLOAD SCHEMA* – provides document + segment anchors
- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – used to generate replay paths
- 📘 *AUDIRA AGENT ONBOARDING FRAMEWORK* – sets up reviewer access rights

---

## 🔷 Section 3: Dashboard Views & Controls

This section defines the **UI-level experience** for authorized reviewers, including what data is visible, what controls are available, and how each agent can be inspected or overridden.

---

### 🖥 Core Views in the Admin Dashboard

| View | Description |
|---|---|
| **Agent Overview Panel** | Summary of agent status, readiness, creator, last update |
| **Readiness Report Viewer** | Full breakdown of FSD_05 validator output |
| **Conflict Resolver Panel** | View, compare, and resolve memory conflicts (FSD_07) |
| **Prompt + Response Simulation** | Run LLM previews using current prompt/memory |
| **Feedback Inbox** | Sortable, filterable feedback queue from users/QA |
| **Override History Log** | List of all past admin interventions per agent |

---

### 🔧 Key Controls & Actions

| Control | Description | Role Permissions |
|---|---|---|

| | | |
|---|---|---|
| **Override Readiness Failure** | Force agent launch with justification | Admin only |
| **Resolve Conflict** | Select correct value or mark for follow-up | Admin + Partner Consultant |
| **Re-run Validator** | Recalculate FSD_05 after update | Admin only |
| **Simulate Agent Response** | Preview how the agent would reply | Admin + QA |
| **Route Feedback to Retraining** | Mark feedback for model update | Admin only |
| **Mark Agent as "Trusted"** | Locks prompt/memory from AI overwrite | Admin only |
| **Flag Segment or Tag** | Mark source content as unreliable or incorrect | Any reviewer |

---

## 📊 Dashboard Filters

| Filter Option | Description |
|---|---|
| **Agent status (e.g., "failed", "ready", "launched")** | |
| **Industry (e.g., retail, SaaS, health)** | |
| **Tag conflict count** | |
| **Last modified (date range)** | |
| **Partner owner** | |
| **"Has unresolved feedback" flag** | |

---

## 🧠 Smart Routing Widgets (Optional AI Boost)

| Widget | Function |
|---|---|
| **Suggest Override Reasons** | LLM recommends justification text based on validator gaps |

| | |
|---|---|
| **Auto-Summarize Feedback Trends** | Group feedback by type, frequency, tag impact |
| **Conflict Resolution Draft Generator** | LLM proposes merged or clarified tag value based on evidence |

---

### 📁 Example View: Agent Readiness Panel

| **Field** | **Value** |
|---|---|
| **Agent ID** | AGENT_00231 |
| **Created By** | partner_consultant_127 |
| **Readiness Score** | 71 (Fail) |
| **Failed Areas** | MISSING_TAG:pricing_model, DOCUMENT_GAPS:overview |
| **Conflict Count** | 2 unresolved |
| **Last Simulated Response** | “Support is 24/7 via Zendesk…” |
| **Override Button** | [Available] |
| **Conflict Panel** | [Resolve Now] |
| **Feedback Linked** | 3 items |

---

📎 **References**:

- 📘 *AUDIRA AGENT BLUEPRINT TEMPLATE* – configures reviewer roles and view permissions
- 📘 *AUDIRA PRE-LAUNCH VALIDATOR SPEC* – feeds report view and override logic
- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – supports simulation and response injection
- 📘 *AUDIRA PRODUCT BLUEPRINT* – defines governance workflow and intervention policies

---

## 🔷 Section 4: Feedback Collection Points

This section outlines where, when, and how structured feedback is collected from users, QA testers, and admins throughout the agent lifecycle.

---

### 🧩 Key Feedback Collection Touchpoints

| Source | Trigger Point | Type of Feedback |
|---|---|---|
| **End User (Live Agent UI)** | After each response or session | “Was this helpful?” thumbs up/down, comment |
| **QA Simulations** | During scripted test runs | Scorecards, tagged errors, simulated corrections |
| **Admin Review Actions** | When resolving conflicts or overriding logic | Comments and audit flags |
| **Partner Consultant Reviews** | Onboarding validation steps | Suggested changes, confidence notes |
| **Post-Deployment Audits** | Periodic quality evaluations | Graded response samples and segment quality scores |

---

### 🗂️ Feedback Categories

| Category | Example |
|---|---|
| **Incorrect Information** | “Agent said we offer 24/7 support, but we don’t.” |
| **Outdated Segment** | “Doc mentions old pricing. New plans launched in 2025.” |
| **Missing Answer** | “Agent didn’t know our return policy.” |
| **Inappropriate Tone** | “Too casual — should be professional in finance use case.” |
| **Hallucination Risk** | “Agent invented info not in memory or prompt.” |
| **Conflicting Memory** | “Two different answers about vendor onboarding.” |

---

### 📝 Feedback Form Fields

Each piece of submitted feedback includes:

```
{
  "feedback_id": "fbk_72819",
  "agent_id": "AGENT_0123",
  "source": "user",
  "feedback_type": "incorrect_info",
  "linked_tag": "payout_policy",
  "linked_response": "We pay vendors every Friday via ACH.",
  "comment": "Not true. We pay on the 15th and 30th via Stripe.",
  "submitted_by": "user_342",
  "timestamp": "2025-06-11T03:49:00Z"
}
```

---

### 🔄 Submission Channels

| Channel | Form Type | Real-time Sync |
|---|---|---|
| **In-agent chat interface** | Pop-up form | ✅ |
| **Admin dashboard override** | Text + tag selector | ✅ |
| **QA console** | Structured test feedback | ✅ |
| **Partner portal** | Free-text + optional prompt view | ✅ |

---

📎 **References**:

- 📘 *AUDIRA AGENT ONBOARDING FRAMEWORK* – identifies who can submit feedback and when
- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – used to link feedback to specific prompt + memory

- 📘 *AUDIRA AGENT SIMULATION TEST KIT* – defines QA flow and feedback capture from scripted tests
- 📘 *AUDIRA FILE & DATA UPLOAD SCHEMA* – maps flagged segments or responses back to document source

---

## 🔷 Section 5: Feedback Processing Logic

This section defines **how feedback is evaluated, categorized, prioritized, and routed** inside the system — enabling both manual triage and automated actions when appropriate.

---

### 🔁 Feedback Intake Pipeline

1. **Feedback Submission**
   → From live user, QA team, admin, or partner.
2. **Auto-Categorization**
   → NLP-based classification + tag detection (e.g., classify as incorrect_info, link to pricing_model).
3. **Confidence Scoring**
   → Based on:
   - Feedback source role (e.g., user vs. admin)
   - Text similarity to memory or prompt
   - Frequency of similar reports for the same tag/segment
4. **Routing Decision**
   → Sent to:
   - Manual triage queue (for critical/conflicting items)
   - Retraining candidate list (if confirmed)
   - Admin reviewer (if tagged as override-worthy)
5. **Linkage Storage**
   → Logs full context:
   - Agent memory at time of issue

- Prompt block used
- Segment or tag involved
- Who submitted the feedback

---

## 🎯 Prioritization Rules

| **Priority Level** | **Trigger Criteria** |
|---|---|
| **High** | Feedback affects core tag, has multiple reports, or comes from QA/admin |
| **Medium** | Single-source feedback with confirmed link to active prompt or memory |
| **Low** | Vague, low-confidence feedback from end users or on optional tags |

---

## 🧭 Example Processed Feedback Object

```
{
  "feedback_id": "fbk_00213",
  "tag_id": "vendor_model",
  "feedback_type": "conflicting_memory",
  "priority": "high",
  "routing_path": ["admin_review", "retraining_queue"],
  "linked_prompt": "AGENT_101_prompt_v4",
  "linked_memory": "vendor_model_v2",
  "status": "open",
  "auto_classification_score": 0.93
}
```

---

## 🛠 System Actions Based on Feedback

| Action | Trigger |
|---|---|
| **Flag Memory for Review** | If feedback contradicts an active memory block |
| **Mark Segment as Outdated** | If flagged for being old, incorrect, or misleading |
| **Soft-Disable Prompt Block** | If associated prompt fragment is triggering confusion |
| **Add to Retraining Set** | If feedback passes trust threshold or is admin-approved |
| **Notify Agent Owner** | For critical issues in live agents |
| **Trigger Follow-up Question** | Sends a new clarification query to the user (via FSD_04) |

---

📎 **References**:

- 📘 *AUDIRA PRE-LAUNCH VALIDATOR SPEC* – defines core vs. optional tag priority
- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – maps feedback to generated prompt structures
- 📘 *AUDIRA AGENT SIMULATION TEST KIT* – used to compare QA feedback against expected response paths
- 📘 *AUDIRA AGENT BLUEPRINT TEMPLATE* – identifies roles with authority to approve feedback-triggered retraining

---

## 🔷 Section 6: Escalation & Triage Logic

This section defines **how flagged issues, feedback, and unresolved memory conflicts are escalated** to the appropriate team or reviewer — ensuring every critical item gets resolved or documented.

---

### 🧭 Escalation Triggers

| Escalation Trigger | Routed To |
|---|---|
| ❗ **Repeated user flags on same tag or response** | QA team + admin reviewer |
| ⚠️ **Memory conflict unresolved after override** | Memory team or AI engineer |

| | |
|---|---|
| 🚫 **Agent answering with outdated or hallucinated info** | Retraining lead + owner |
| 🛑 **Failed validator override with risk tags** | Executive reviewer with justification log |
| ❓ **Feedback on compliance, legal, or health tags** | Legal/partner compliance team |
| 🔄 **Ambiguous multi-source segment disagreement** | Sent to document integrity triage queue |

---

### 🗂️ Issue Routing Logic

Each flagged item is assigned:

- issue_type: e.g., conflict, hallucination, outdated info
- severity_level: auto-detected or reviewer-assigned
- assigned_to: team or individual based on tag/category ownership
- status: open / under review / resolved / retraining scheduled
- escalation_trace: full record of who handled what and when

---

### 📈 Escalation Panel in Dashboard

| Field | Description |
|---|---|
| **Issue Type** | Memory conflict, bad response, prompt mismatch |
| **Affected Tags** | One or more discovery tags linked |
| **Affected Agent** | ID, owner, launch date |
| **Last Action** | E.g., “Manually resolved by QA” or “Pending retraining” |
| **Days Open** | SLA tracking for unresolved issues |
| **Status** | Open, Assigned, Resolved, Archived |

---

### 🔁 Auto-Loop to Retraining or Validator

If resolution involves factual change:

- Agent memory is patched (FSD_07)
- Prompt chain may be recompiled (FSD_06)
- Readiness validator is optionally re-run (FSD_05)

For major issues (e.g., hallucination from prompt logic), the escalation may trigger a **prompt chain logic audit**.

---

### 🔒 Audit Trail Requirements

Each escalated case must maintain:

- Feedback text or source trace
- Action history (resolves, rejections, reassignment)
- If override used: justification + reviewer ID
- Snapshot of memory and prompt at time of issue

---

📎 **References**:

- 📘 *AUDIRA PRODUCT BLUEPRINT* – defines escalation thresholds and reviewer responsibilities
- 📘 *AUDIRA AGENT BLUEPRINT TEMPLATE* – maps agents to reviewers, QA teams, and risk level
- 📘 *AUDIRA FILE & DATA UPLOAD SCHEMA* – allows document-linked triage and escalation
- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – provides structure for audit snapshot reconstruction

---

## 🔷 Section 7: Retraining Hooks

This section defines **how validated feedback, memory updates, and escalated cases are used to trigger targeted retraining** of LLM agents — ensuring continuous improvement and alignment with each SMB's evolving profile.

---

### 🔁 What Can Trigger Retraining?

| Trigger Type | Description |
|---|---|
| ✅ **Admin-confirmed feedback** | Reviewed and approved correction or clarification |
| ✅ **Overridden memory conflict** | Once a contradiction is resolved and marked as "final" |
| ✅ **High-volume user feedback** | Same tag or response flagged by multiple users or sessions |
| ✅ **QA flagged test failure** | Tagged with retraining suggestion during simulation |
| ✅ **Prompt-to-response mismatch** | Validated case of prompt block resulting in hallucination |
| ✅ **Agent delta after re-onboarding** | When documents or answers change post-launch |

---

### 📦 Retraining Data Format

All training candidates are compiled as TrainingSample objects:

```
{
  "agent_id": "AGENT_1021",
  "issue_type": "conflicting_memory",
  "corrected_tag": "payout_policy",
  "correct_value": "Automated via Stripe every 14 days",
  "source": "admin_resolution",
  "linked_prompt": "prompt_v5",
  "llm_response": "Manual payout confirmed",
  "training_action": "negative_sample",
  "timestamp": "2025-06-11T04:00:00Z"
}
```

---

## 🧠 Training Actions

| **Action Type** | **Usage** |
|---|---|
| **Positive Sample** | “This is what the agent should say in this context.” |
| **Negative Sample** | “This is what the agent said incorrectly — avoid this.” |
| **Replacement Sample** | “Replace this behavior with updated memory or tone.” |

---

## 📁 Retraining Hooks Queue

| **Queue** | **Triggered By** | **Processing Mode** |
|---|---|---|
| **training/live_corrections** | Resolved feedback from users/admin | Batched nightly |
| **training/conflict_resolution** | Memory contradictions resolved | Continuous |
| **training/qa_failures** | Failed simulations | Weekly model update round |
| **training/hallucination_cases** | Prompt logic overrides | Priority retrain window |
| **training/new_tags** | New tag types or values not yet modeled | Tag dictionary + prompt template updates |

---

## 🧪 Evaluation Before Push

Before integrating into model fine-tuning:

- Each sample is evaluated for clarity and source traceability
- Language and tone are normalized

- Examples are tested in **agent simulation preview mode** (FSD_10)
- Approved samples are added to the training corpus per agent class (e.g., "Retail SME agent v3.2")

---

📎 **References**:

- 📘 *AUDIRA AGENT SIMULATION TEST KIT* – used for retraining sample QA before LLM fine-tuning
- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – needed for prompt-response alignment during model update
- 📘 *AUDIRA AGENT ONBOARDING FRAMEWORK* – defines class-level agent groupings for training sets
- 📘 *AUDIRA DISCOVERY TAGS DICTIONARY* – expanded when training introduces new or rare tag patterns

---

## 🔷 Section 8: Future Enhancements

This section outlines strategic improvements and tooling extensions planned to evolve the dashboard and feedback loop into a full **QA governance hub** for Audira and its partners.

---

### 🚀 Upcoming Features & Enhancements

| Feature | Description | Benefit |
|---|---|---|
| **Real-Time Feedback Streaming** | View live thumbs up/down and user comments as they happen | Enables instant QA visibility |
| **Audit Trail Explorer** | View full lifecycle of memory/prompt changes, overrides, and response deltas | Improves traceability and compliance |
| **Feedback Heatmaps** | Visualize which tags or topics are triggering the most flags or corrections | Identifies knowledge gaps at scale |

| | | |
|---|---|---|
| **Auto-Labeling with LLMs** | Use AI to pre-label feedback and classify root causes | Saves human triage time |
| **Reviewer Scoring Dashboard** | Tracks reviewer activity, resolution times, accuracy of overrides | Enables internal performance metrics |
| **Agent Benchmark Simulator** | Periodic test harness for launched agents, scored against a standard QA set | Enables automated regression testing |
| **Time-Sensitive Memory Reminders** | Prompts admins to review memory blocks flagged as stale or policy-expired | Improves memory freshness |
| **Custom QA Templates by Industry** | Tailor review forms and escalation rules per vertical (e.g., fintech vs. health) | Enhances reviewer relevance |
| **Feedback Reuse Suggestions** | Surface similar past issues and resolution paths when a new feedback case matches prior ones | Boosts reviewer efficiency |

---

## 🧰 Compatible OSS Tools

| Tool | Role |
|---|---|
| **TruLens / DeepEval** | Feedback scoring and fine-tuning evaluation |
| **LangGraph** | Issue triage routing and override workflows |
| **PromptLayer / PromptTools** | Prompt-response audit versioning |
| **Label Studio** | Optional front-end for feedback classification |
| **Weights & Biases** | Logging model changes triggered by dashboard feedback |
| **Unstructured.io** | Segment mapping for flagged source fragments |

---

📎 **Linked Modules**:

- 📘 *AUDIRA PRODUCT BLUEPRINT* – sets roadmap for feedback-to-training evolution
- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – enables response-to-feedback mapping

- 📘 *AUDIRA PRE-LAUNCH VALIDATOR SPEC* – source of override, scoring, and readiness data
- 📘 *AUDIRA AGENT BLUEPRINT TEMPLATE* – defines reviewer roles, escalation privileges, and agent groupings

---