Training Dashboard for Mavericks Onboarding

Problem Statement

Onboarding and training new Mavericks involves a comprehensive two-phase program: Foundation and Role-Specific Training. The Foundation program is designed to provide a holistic learning experience, covering functional, behavioral, and technical skills through experiential and business-oriented project-based learning methodologies, enhanced by gamification.

Key challenges in managing this training process include:

- **Multiple Touch Points:** Ensuring consistent communication and support to address challenges and improve the overall program.
- **Structured Learning Schedule:** Managing a daily learning schedule that includes quizzes, coding challenges, and project-based assessments.
- **Assessment Tracking:** Monitoring various assessment scores such as coding, assignments, learning completion, and external certification completion.
- **Profile Management:** Updating and maintaining the complete profile of each fresher, accessible to both candidates and administrators for analysis.
- **Reporting:** Generating downloadable reports for comprehensive analysis and record-keeping.

This solution aims to streamline the training process, ensuring that all aspects of the program are efficiently managed and tracked.

Expected Solution Overview

We will leverage an open-source AI-based multi-agent framework to decompose the workflow into specialized, cooperating agents, and expose control and status via a rich Web UI. Below are examples of the agents involved:

Agent	Role & Inputs	Outputs	Reasoning / Generative AI Use
Onboarding Agent	Fresher profiles, training schedules	Structured onboarding plans	Create personalized onboarding plans based on fresher profiles and training schedules.
Assessment Agent	Quiz results, coding challenge scores, assignment submissions	Assessment scores and feedback	Evaluate and provide feedback on various assessments to track progress.
Profile Agent	Assessment scores, certification completions, learning progress	Updated fresher profiles	Maintain and update fresher profiles with the latest

			assessment scores and achievements.
Reporting Agent	Assessment data, profile updates	Downloadable reports	Generate comprehensive reports for analysis and record-keeping.

We can use any database and synthetic data to simulate the solutions.

Web UI Design

This solution must provide both fresher candidates and administrators with web UI pages with live visibility.

For Fresher Candidates:

- **Current Training Status Dashboard:**
  - Daily Quiz Status (Completed / Pending)
  - Coding Challenge Progress (Not Started / In Progress / Completed)
  - Assignment Submissions (Submitted / Pending)
  - Certification Completion (Completed / In Progress)
- **Real-Time Workflow Progress Bar with below action items:**
  - Profile Updated
  - Daily Quiz Completed
  - Coding Challenge Submitted
  - Assignment Submitted
  - Certification Completed

For Administrators:

- **Admin Console includes:**
  - Fresher Search & Filters (by skill, department, status).
  - View agentic framework-managed queues, latencies, error rates.
  - Report Generation about training progress by individual or department.

This design ensures that both fresher candidates and administrators have real-time visibility and control over the training process, leading to more efficient and effective onboarding and skill development.

Document Ingestion & Classification

Problem Statement

Enterprises generate hundreds of thousands of unstructured documents daily—PDFs, Word files, scanned images, emails. Manual sorting and routing is:

- **Slow & Costly:** Human reviewers become a bottleneck.
- **Error-Prone:** Mis-tagged contracts slip through compliance gaps.
- **Unscalable:** Growing volume overwhelms static workflows.

Goal: Build an AI-powered MAS that automatically ingests, extracts, classifies, and routes documents at scale—reducing manual work, cutting errors, and providing full visibility on processing status.

2. Solution Overview

We decompose the pipeline into four collaborating agents on the opensource agentic framework. Each agent reasons over its inputs, may leverage Generative AI/LLMs for hard NLP tasks, and emits structured events. A central Message Bus connects them; stateful agents track progress per document. Here we are give example agents and its reasoning.

Agent	Inputs	Outputs	Reasoning / Generative AI Use
Ingestor Agent	File-share events, email hooks, upload API calls	Raw document payload + metadata event (doc.received)	• **Reasoning:** Decide priority based on sender, file size, or folder. • **LLM-Optional:** Summarize email bodies to extract subject/context.
Extractor Agent	doc.received event; document binary	Text + structured entities (doc.text, doc.entities)	• **OCR/NLP:** Use OCR engines → LLM to clean up noisy text. • **Entity Extraction:** LLM prompts to pull dates, parties, amounts.
Classifier Agent	Extracted text & entities	Document type label + confidence (doc.type, score)	• **Model Inference:** Fine-tuned classifier (or LLM zero-shot) to tag invoice/contract/etc. • **Reasoning:** If low confidence, flag for human review.
Router Agent	Classified document event (doc.type)	Delivery commands (ERP API call, DMS upload, email alert)	• **Routing Logic:** Map types → systems (e.g. invoice → accounting ERP). • **Fallback Reasoning:** If target system down, enqueue retry or send Slack alert.

3. Web UI Design

A webUI gives both operators and admins live visibility into every document's journey:

3.1 Operator Dashboard

1. **Upload/Ingestion Panel**
    - Drag-and-drop or "Connect Mailbox" button.
2. **Workflow Progress Bar**
    - Nodes:
        1. **Ingested**
        2. **Extracted**
        3. **Classified**
        4. **Routed**
    - Each lights up with timestamp as agents complete their steps.
    - Hover reveals details (e.g. "OCR took 3.2s," "Type=Contract (94% confidence)").
3. **Document List & Status**
    - Table of recent docs with columns: Name, Type, Status, Last Updated.
    - Click row to expand the live progress bar and metadata panel.
4. **Manual Overrides**

Buttons to "Re-extract," "Re-classify," or "Route to…" if human intervention is needed

Document Similarity Comparison and Ranking

Problem Statement

In the recruitment process, it is essential to efficiently match job descriptions (JDs) with the profiles of available consultants. This involves comparing and ranking the similarity between two types of text documents:

- **Document #1:** Job Description (JD) embedded in an AR (Automated Requisition).
- **Document #2:** Set of profiles of consultants.

The goal is to compare the similarity of these documents based on skills, experience, and contextual information, and then rank the consultant profiles based on their match with the JD. The system should send an email to the AR requestor with the top 3 matches or notify the recruiter if no suitable matches are found.

Key challenges include:

- **Contextual Comparison:** Accurately comparing documents based on skills, experience, and contextual relevance.
- **Ranking Profiles:** Effectively ranking consultant profiles based on their similarity to the JD.
- **Automated Communication:** Sending automated emails to the AR requestor or recruiter based on the comparison results.

This solution aims to streamline the recruitment process by automating the comparison and ranking of consultant profiles against JDs, ensuring efficient and accurate matching.

Expected Solution Overview

We will leverage an open-source AI-based multi-agent framework to decompose the workflow into specialized, cooperating agents, and expose control and status via a rich Web UI. Below are examples of the agents involved:

Agent	Role & Inputs	Outputs	Reasoning / Generative AI Use
Comparison Agent	Job Description (JD), Consultant Profiles	Similarity scores between JD and profiles	Compare documents based on skills, experience, and contextual information.
Ranking Agent	Similarity scores	Ranked list of consultant profiles	Rank profiles based on their similarity scores to the JD.

Communication Agent	Ranked list of profiles, AR requestor email, Recruiter email	Automated emails to AR requestor or recruiter	Send emails with top 3 matches or notify the recruiter if no matches are found.

We can use any database and synthetic data to simulate the solutions.

Web UI Design

This solution must provide both recruiters and AR requestors with web UI pages with live visibility.

For AR Requestors:

- **Current Matching Status Dashboard:**
  - JD Comparison Status (Completed / In Progress)
  - Top 3 Matches (Listed / Not Found)
  - Email Notification Status (Sent / Pending)
- **Real-Time Workflow Progress Bar with below action items:**
  - JD Compared
  - Profiles Ranked
  - Email Sent to AR Requestor

For Recruiters:

- **Admin Console includes:**
  - JD Search & Filters (by skills, experience, status).
  - View agentic framework-managed queues, latencies, error rates.
  - Report Generation about matching results by JD or consultant profile.

This design ensures that both AR requestors and recruiters have real-time visibility and control over the document comparison and ranking process, leading to more efficient and effective recruitment.

Mavericks Coding Platform

Problem Statement

Developers and companies face significant challenges in skill certification, interview preparation, and continuous learning. Traditional methods of skill assessment and learning are:

- **Inefficient**: Manual review of coding skills is time-consuming and resource-intensive.
- **Inconsistent**: Subjective evaluations can lead to biased skill assessments.
- **Unscalable**: Growing demand for skilled developers overwhelms traditional learning workflows.
- **Lack of Engagement**: Developers need motivation and recognition for their learning efforts.

Goal: Build a comprehensive coding platform that automates the assessment of coding skills, provides personalized learning paths, facilitates engaging hackathons, and offers full visibility into user progress and performance. The platform should be gamified with leaderboards and reports to enhance user engagement and provide actionable insights.

Solution Overview

We decompose the platform into several collaborating agents within an open-source agentic framework. Each agent processes its inputs, leverages AI and machine learning for complex tasks, and emits structured events. A central message bus connects these agents, and stateful agents track the progress of each user. Below are examples of agents and their reasoning:

Agent	Inputs	Outputs	Reasoning / Generative AI Use
Profile Agent	User registration data, past performance, resume uploads	User profile + skill vectors (user.profile)	• Reasoning: Assess user skills based on past performance and uploaded resumes. • AI-Optional: Use NLP to extract and summarize key skills and experiences from resumes.
Assessment Agent	User profile, quiz/exercise triggers	Proficiency scores across competencies (user.scores)	• AI/ML: Dynamically select quiz difficulty based on real-time performance. • Reasoning: Adjust

			assessment difficulty based on user responses.
Recommender Agent	User profile + assessment scores + content metadata	Personalized learning path (user.learning_path)	• Model Inference: Recommend learning modules based on skill gaps. • Reasoning: Provide narrative explanations for recommended modules.
Tracker Agent	Learning Management System (LMS) API events, quiz re-scores	Progress updates and alerts (user.progress)	• Reasoning: Analyze completion patterns to detect user progress or stagnation. • AI-Optional: Trigger additional assessments or refreshers if stagnation is detected.
Hackathon Agent	Hackathon setup requests, challenge definitions	Hackathon events and challenges (hackathon.setup)	• Reasoning: Facilitate the setup and management of hackathons. • AI-Optional: Use AI to generate real-world challenges and evaluate submissions.

Web UI Design

A web UI provides both users and administrators with live visibility into every step of the assessment, learning, and hackathon process:

User Dashboard

1. **Assessment Panel**
    - Upload or connect coding exercises.
    - View upcoming and past assessments.
2. **Progress Bar**
    - Nodes:
        1. Profile Created
        2. Assessment Completed
        3. Skills Evaluated

    4. Learning Path Generated
- Each node lights up with a timestamp as agents complete their steps.
- Hover reveals details (e.g., “Assessment took 15 minutes,” “Score=85%”).

3. **Learning Path & Status**
    - Table of recommended learning modules with columns: Module Name, Estimated Time, Completion Status.
    - Click row to expand details and start learning.
4. **Manual Overrides**
    - Buttons to “Re-assess,” “Update Profile,” or “Request Review” if manual intervention is needed.
5. **Hackathon Panel**
    - Join or create hackathons.
    - View hackathon challenges and submissions.
6. **Leaderboard & Achievements**
    - View leaderboards and earned badges.
    - Track progress and achievements.

Admin Dashboard

1. **User Search & Filters**
    - Search and filter users by skill, score, or learning path.
2. **Workflow Progress Bar**
    - Nodes:
        1. Profile Loaded
        2. Assessment Completed
        3. Skills Evaluated
        4. Learning Path Generated
    - Each node lights up with a timestamp as agents complete their steps.
    - Hover reveals details (e.g., “Assessment took 15 minutes,” “Score=85%”).
3. **User List & Status**
    - Table of users with columns: Name, Skills, Assessment Score, Last Updated.
    - Click row to expand live progress bar and metadata panel.

4. **Manual Overrides**
    - Buttons to “Re-assess,” “Update Profile,” or “Generate Report” if manual intervention is needed.
    - **Hackathon Management**
    - Create and manage hackathons.
    - View hackathon submissions and results.
7. **Reports & Analytics**
    - Generate reports on user progress and platform usage.
    - View analytics on skill development and engagement.

Personalized Learning & Skill-Gap Analysis

Problem Statement

Large enterprises struggle to deliver training that truly matches each employee's existing skills and evolving gaps. Traditional LMS platforms offer "one-size-fits-all" curricula or simple role-based tracks, leading to:

- Low engagement: Employees see irrelevant content.
- Inefficient spending: Budgets go to courses that aren't needed.
- Stagnant skill growth: Hidden gaps remain unaddressed.

In Hexaware learning is based on the TSR role of the consultant. This solution evaluate the consultant current profile/skill against the TSR skill and find the Gaps of learning, guide the consultant for learning.

Expected Solution Overview

We'll leverage any opensource AI based multi-agent framework to decompose the workflow into specialized, cooperating agents, and expose control and status via a rich Web UI. Example on the Agents are given below.

Agent	Role & Inputs	Outputs	Reasoning
Profile Agent	HR/ERP data, past course completions, performance ratings	Employee skill vectors & history logs	Infer latent skill vectors by correlating disparate data (e.g. past courses, KPI scores).
Assessment Agent	Triggers quizzes/exercises; collects answers and timing	Proficiency scores across competencies	Dynamically select difficulty levels based on real-time quiz performance (e.g. "If they ace questions on Topic A, ramp up Topic B").
Recommender Agent	Profiles + assessment scores + content metadata	Ranked learning path	Craft narrative explanations ("Why this module matters for your Java concurrency gap") and write personalized learning-path descriptions.
Tracker Agent	LMS API events (course started/completed), quiz re-scores		Analyze completion patterns and quiz re-scores to detect if a user is plateauing, then trigger a mini-assessment or refresher.

We can use any database and synthetic data to simulate the solutions.

Web UI Design

This solution must provide both learners and administrators webUI pages with live visibility.

- Current Learning Path for Learners (Title, estimated time, completion status (Not Started / In Progress / Done))
- Real-Time Workflow Progress Bar with below action items for Learners
    a. Profile Loaded,
    b. Assessment Pending,
    c. Assessment Completed,
    d. Recommendations Generated,
    e. Learning In Progress
- Admin Console includes
    1. User Search & Filters (by skill gap, department).
    2. view agentic framework-managed queues, latencies, error rates.
    3. Report Generation about learning by individual or department

Pool Consultant Management System

Managing a pool of consultants efficiently is a complex task that involves tracking various aspects of their professional development and engagement. Traditional methods often lead to inefficiencies such as:

- **Outdated Information:** Resumes and skill sets are not regularly updated.
- **Poor Attendance Tracking:** Difficulty in monitoring and reporting attendance for bench consultant meetings.
- **Lack of Opportunity Insight:** Insufficient documentation of opportunities provided during bench periods.
- **Training Gaps:** Inadequate tracking of training sessions and certifications completed.

In Hexaware, consultants are often on the bench waiting for project allocations. This solution aims to streamline the management of consultants by automating the collection and reporting of essential data, ensuring that consultants are well-prepared and efficiently utilized.

Expected Solution Overview

We will leverage an open-source AI-based multi-agent framework to decompose the workflow into specialized, cooperating agents, and expose control and status via a rich Web UI. Below are examples of the agents involved:

Agent	Role & Inputs	Outputs	Reasoning / Generative AI Use
Resume Agent	Latest resumes from consultants	Updated skill vectors & history logs	Infer skill vectors by analyzing resume updates and past experiences.
Attendance Agent	Meeting attendance data from Teams	Consolidated attendance reports	Correlate attendance data to generate comprehensive reports.
Opportunity Agent	Opportunities provided during bench period	Number of opportunities documented	Track and document opportunities to ensure consultants are engaged.
Training Agent	Training sessions and certifications completed	Training and certification reports	Record and analyze training data to identify skill gaps and growth.

We can use any database and synthetic data to simulate the solutions.

Web UI Design

This solution must provide both consultants and administrators with web UI pages with live visibility.

For Consultants:

- **Current Status Dashboard:**
  - Resume Update Status (Updated / Pending)
  - Attendance Report (Completed / Missed)
  - Opportunities Provided (Number of Opportunities)
  - Training Progress (Not Started / In Progress / Completed)
- **Real-Time Workflow Progress Bar with below action items:**
  - Resume Updated
  - Attendance Reported
  - Opportunities Documented
  - Training Completed

For Administrators:

- **Admin Console includes:**
  - Consultant Search & Filters (by skill, department, status).
  - View agentic framework-managed queues, latencies, error rates.
  - Report Generation about consultant status by individual or department.

This design ensures that both consultants and administrators have real-time visibility and control over the consultant management process, leading to more efficient and effective utilization of resources.