Project ID: 28

Project Title

Ai Agent Conductive Feedback Loop

Client Name

Lamont Tang

Group Capacity

4 groups

Project Background

Project Background

AI Agent Conductive Feedback Loop (CFL) is a ~ 8 week research and development initiative aimed at creating a dynamic, rubric-driven platform for evaluating and ranking AI workflows. Designed to support executive and stakeholder decision-making, CFL helps surface high-impact AI use cases by visualizing workflow performance and feedback data in an intuitive, modular dashboard. The system emphasizes minimal resource expenditure while maximizing strategic insights through intelligent UI, human-centered design, and real-time feedback mechanisms.

The project also explores emerging concepts like visual performance artifacts and trust indicators, setting the foundation for future systems that support reputation-building and skill signaling in the agentic AI ecosystem. Students will work on key interface and logic components to enable these features, while proprietary algorithms and data weighting models will remain abstracted from scope.

Project Goals

- Build a modular, insight-rich dashboard that translates user–AI interactions into clear, actionable visualizations.
- Establish a rubric-based scoring and feedback system that enables benchmarking, behavior nudges, and self-improvement loops.
- Deliver a polished, production-ready front end with personalized agent performance summaries, visual diffing of prompts, and embedded documentation for executive clarity.
- Enable comparative analysis (“You vs Past” and “You vs Peer”) to encourage reflection and performance gains.

- Enable rapid decision-making through visualized AI performance insights
- Create low-friction feedback loops that help users improve over time

- Prepare system foundations for future integration of trust and reputation metrics

- Train a team in CS and HCI best practices, including temporal modeling, declarative UI design, feedback loop integration, and performance-aware front-end deployment.

Project Scope

Project Scope

The AI Agent Conductive Feedback Loop (CFL) project will deliver a web-based platform that captures, evaluates, and visualizes user–AI interactions to support executive decision-making and workflow optimization. The project is structured into three agile sprints over 10 weeks and focuses on the following scope areas:

In Scope

1. UI/UX Research & Design

- Competitive analysis of similar tools (e.g., Relevance AI, Relay.app, n8n)
- Figma-based wireframes (low and high fidelity)
- Establishment of a component-first design system

2. Rubric-Driven Scoring System

- Design and implementation of a feedback rubric for AI workflows
- Visualization of rubric breakdown, score explanations, and performance insights
- Comparative metrics: “You vs Past” and “You vs Peer”

3. Interaction Modeling & Insight Cards

- Logging and schema design for user–agent interactions
- Session playback, prompt diffing, and progress trendlines
- Insight cards with documentation, behavior nudges, and media

4. Front-End Development

- Modular dashboard with interactive filtering and responsive UI
- State management and data-binding logic
- Integration of mock backend APIs and YouTube embeds

5. Usability Testing & Deployment

- Final polish and performance optimization
- Accessibility checks and cognitive load reduction strategies
- Deployment to production (e.g., Vercel)

6. Learning & Documentation

- Embedded user documentation and walkthrough videos
- Capture and share learnings for future student teams

Out of Scope

- Full production-grade backend (e.g., real-time API scalability, auth, data storage)
- Proprietary LLM integration or training (mock interaction data will be used)
- Extensive mobile or desktop-native app support (web-first only)
- Long-term maintenance post-deployment

Project Requirements

Project Requirements

1. Interaction Modeling & Data Logging

Purpose: Capture structured records of how users interact with AI workflows to enable analysis and feedback.

- Track user inputs, agent responses, timestamps, and outcome ratings.
- Normalize data into a temporal schema for session replay and comparison.
- Store interaction history to support session-based evaluation and progress tracking.

2. Modular UI Framework

Purpose: Enable scalable, maintainable UI development across insight components.

- Component-first architecture (e.g., buttons, cards, panels, charts).
- Consistent design system (typography, color palette, layout grid).
- Responsive design across standard screen sizes.
- Front-end built with TypeScript/React or similar framework.

3. Rubric-Based Scoring System

Purpose: Provide structured evaluation of user workflows for feedback and comparison.

- Predefined rubric with multiple criteria (e.g., time-to-value, prompt clarity, reuse potential).

- Weighting of rubric elements per use case or user level.
- Real-time score generation from interaction metadata.

4. Insight Cards & Feedback Delivery

Purpose: Offer interpretable insights based on rubric scores and session metadata.

- Dynamic cards containing:
- Score explanation (e.g., "Your clarity score dropped 10% vs last session")
- Behavioral nudges ("Try simplifying your input prompts")
- Embedded support media (docs, YouTube tutorials, etc.)
- Clickable links to related documentation or peer workflows.

5. Comparative & Trend Analysis

Purpose: Allow users to see improvement or identify regressions over time or relative to peers.

- "You vs Your Past Self" comparison view
- Timeline of score deltas across sessions
- Highlighted progress and regression zones
- "You vs Peer Benchmark"
- Anonymized or pseudo-anonymized comparison
- Median, top-performer, or cohort views

6. Scoring Dashboard

Purpose: Aggregate insights into a high-level, decision-friendly interface.

- Interactive filters by session, rubric dimension, or workflow tag
- Summary panels with average scores, deltas, and recent highlights
- Personalized feedback loops that adapt based on performance trends

7. Session Playback & Prompt Diffing

Purpose: Help users analyze how their inputs and workflow choices evolve.

- Timeline-based replay of input–output sequences
- Highlighted changes in prompts and agent responses
- Optional tooltips explaining impact of prompt changes on scoring

8. Usability & Testing

Purpose: Ensure that the platform is accessible, intuitive, and low-friction.

- Heuristic usability testing with sample users
- Cognitive load minimization (visual clarity, progress markers, tooltips)
- Accessibility checks (WCAG compliance for color, text, and interaction)
- Final deployment to a cloud hosting platform (e.g., Vercel)

9. Documentation & Deployment Tools

Purpose: Support onboarding and ongoing usage of the platform.

- Integrated documentation assistant (FAQs, video embeds)
- Admin UI for uploading rubric changes or walkthroughs
- Embed mock YouTube API to show video tutorials and view counts
- Trust signals like upvotes, remix counts, and usage stats

Required Skills

To successfully contribute to the development of the AI Agent Conductive Feedback Loop (CFL) platform, students should possess or be willing to rapidly develop the following technical and analytical competencies:

Frontend Development

- HTML/CSS/JavaScript fundamentals
- TypeScript for typed, scalable component development
- React.js or similar frontend frameworks for building modular UIs
- Experience with Figma for translating wireframes into working interfaces

Computer Science Fundamentals

- Data structures and algorithms relevant to UI state, time-series, and interactivity
- Understanding of schema normalization and interface contracts
- Concepts of declarative UI design and state synchronization
- Exposure to temporal modeling of interactions and logging systems

Data Visualization & UX Design

- Familiarity with tools/libraries for rendering visualizations (e.g., D3.js, Chart.js)
- Understanding of information hierarchy and human–computer interaction (HCI) principles

- Ability to create clean, interpretable dashboards and score breakdowns

Testing & Deployment

- Basic usability testing techniques (e.g., heuristic evaluation, A/B testing)
- Knowledge of frontend performance optimization and accessibility standards
- Exposure to CI/CD tools and cloud deployment platforms (e.g., Vercel)

Feedback Loop Design

- Familiarity with feedback mechanisms in learning or system optimization
- Experience or interest in behavioral nudges, scoring rubrics, or recommendation systems

Soft Skills & Team Collaboration

- Strong communication skills to articulate ideas and collaborate on UI/UX decisions
- Ability to work in an agile, sprint-based environment
- Willingness to engage in peer review, design critique, and feedback iteration

Expected Outcomes

1. Working Front-End Dashboard
    - Rubric panels, insight cards, filters, timeline replay
    - Mock data and scoring logic integrated
2. Figma Design System & Wireframes
3. Documentation
    - Developer guide
    - User guide
    - Technical system overview
4. Walkthrough Demo & Usability Report
    - Video or slide walkthrough
    - Feedback summary and improvements
5. Simulated Trust Signals (Optional)
    - Basic scoring badges, view counts, and remix metadata (using mock data)
    - UI mockups for “visual performance snapshots” (abstracted ProofTiles)

Disciplines

Software Development;Web Application Development;Artificial Intelligence (Machine/Deep Learning, NLP);Generative AI (GenAI);Computer Science and Algorithms;Big data Analytics and Visualization;Human Computer Interaction (HCI);Cloud Computing;

Other Resources

To support student success and ensure smooth development, the following tools, APIs, and reference materials will be made available during the project:

Development Infrastructure

- Version Control:

GitHub repository access (private or team-specific forks) with pre-set branches and contribution guidelines.

- Deployment Platform:

Access to Vercel for frontend hosting and continuous deployment.

- Mock Backend / APIs:
- Sample JSON datasets for interaction logs, rubric scores, and session metadata
- Mock API endpoints using MockAPI.io or JSON Server
- Optional YouTube-like API schema for embedding documentation content (title, views, trust signals)
- Cloud Tools (Optional):
- Temporary access to Replit, Codesandbox, or Render for staging and prototyping
- Shared Figma workspace for UI/UX collaboration

Technical Starter Kits & Templates

- Pre-configured React + TypeScript project template
- Prebuilt component scaffolds for cards, filters, charts, and panels
- Rubric definition template (editable JSON/YAML format)
- Insight Card structure guide (how to map scores to recommendations/media)

Design Resources

- Figma Component Library with core visual styles, icons, and layout grids
- Competitive UI/UX audit slides (Relay.app, Relevance AI, etc.)
- Style guidelines for dashboard clarity, data visualization, and accessibility

Reference Materials

- Recommended reading on feedback loops and interaction design:
- “The Design of Everyday Things” by Don Norman
- Nielsen Norman Group’s articles on usability heuristics
- Articles on rubric-based assessment in AI

- Documentation Examples:
- Agent feedback examples from GitHub Copilot Labs, Replit Ghostwriter
- UI examples from Amplitude, Notion, Linear