

Data Strata: “Code the Beat” – AI Music Hackathon

Where Algorithms Find Their Rhythm.

About the Event

Data Strata: “Code the Beat” is a unique AI Music Hackathon that brings together creativity, technology, and rhythm.

Participants will explore how **machine learning, data, and sound** can harmonize to create interactive, expressive, and story-driven projects.

From **AI-generated melodies** and **emotion-aware soundscapes** to **data-driven visualizations** and **interactive music tools**, this event celebrates innovation through rhythm and imagination.

Let your algorithms groove and your ideas make some noise!

Event Flow

Round 1 – Online Prototype Round

Problem Statement Release: *09th January 2026, 6:00 PM*

Submission Deadline: *16th January 2026, 6:00 PM*

Overview:

Teams will receive a broad theme or problem statement to spark creativity. Participants must build a **prototype** that reflects their idea — it can be experimental, minimal, or a complete proof-of-concept.

What to Submit:

- A **demo video (2 minutes)** showcasing the project.
 - Treat it like a *product launch pitch* — present your idea, show the interface, and convey your creativity.
- A **short project description** (via the provided form).
- (Optional) **GitHub or deployed link** if applicable.

Evaluation:

Projects will be assessed on creativity, relevance, and clarity of concept. Selected teams will qualify for the next round.

Round 2 – Offline Presentation Round

Date: 17th January 2026

Mode: Offline (Venue to be announced)

Overview:

Shortlisted teams will present their projects in an **offline PPT presentation round** before a panel of judges.

Your presentation should cover:

- Concept and inspiration behind your project
- Demonstration of your prototype (if available)
- Technical workflow and implementation
- Challenges faced during development
- Future improvements and scope

The focus will be on originality, clarity, and creativity of execution.

Problem Statement

Design and develop an AI-powered music or audio system that uses data and machine learning to generate, analyze, transform, or interact with sound in a meaningful way. The system must represent a clear story, mood, or concept (such as emotion-driven compositions, narrative-based music evolution, or interactive audio systems) and demonstrate both creative originality and technical clarity.

Participants are expected to go beyond simple model usage and show thoughtful system design, transparent methodology, and a clear connection between the technical approach and the intended creative outcome.

Technical and Creative Constraints

Library Usage: Open-source tools and pre-trained models are allowed with proper credit. Teams must demonstrate a unique implementation, extension, or wrapper around any pre-trained model used.

Audio Processing: Projects should use specialized audio libraries such as Librosa, Magenta, Essentia, or pydub for analysis or generation.

Architecture: Teams must be able to clearly explain their data sources, preprocessing

steps, model architecture, and system workflow.

Thematic Alignment: Each project must represent a specific story, mood, or concept and reflect it through the system's behavior and output.

Model Lineage C Attribution: If pre-trained models are used, teams must document fine-tuning details or prompt engineering strategies and explain how they affect the results.

Guidelines

General Rules

- **Team Size:** Up to 3 *members*
- **Eligibility:** Open to all
- **Originality:** All work must be original. You may use open-source tools or pre-trained models with proper credit.

Project Theme Ideas (For Inspiration)

- AI that composes music from text prompts or genre selections – where users describe a mood, style, or theme, and the system creates matching music.
- Prompt-based remixing or blending tools – generate new versions of existing tracks guided by creative instructions.
- Sound-to-visual experiences – create visuals, animations, or patterns that react to AI-generated music.
- Interactive music systems – where users influence rhythm, melody, or instruments in real time through prompts or actions.
- Storytelling through AI music – turn textual or narrative prompts into musical compositions that evolve with the story

Technical Guidelines

- Any programming language or framework may be used (Python preferred).
- Recommended ML libraries: **TensorFlow, PyTorch, scikit-learn**.
- For audio/music projects: **Librosa, Magenta, pydub, Essentia** are great choices.
- If creating an app or interface:
 - **Frontend:** HTML/CSS, React, or Streamlit
 - **Backend:** Flask, FastAPI, or Node.js
- Use **open-source datasets** (e.g., GTZAN, MAESTRO) or collect your own small dataset.

UI / UX Guidelines

- Keep the design **simple, clean, and intuitive**.
- The interface should complement your project's **theme or mood** – bright and playful for upbeat music, calm and minimal for ambient sounds.
- Prioritize **clarity and storytelling** – ensure users or judges can easily grasp what your project does.
- A small, well-thought-out UI is better than a complex, cluttered one.

Music C Creativity Guidelines

- The generated or analyzed music should **represent a story, mood, or concept**.
- You can base your project on:
 - *Emotion-driven compositions*
 - *Music transformations* (e.g., remixing or blending based on data)
 - *Interactive or responsive music systems*
- Focus on **creative originality over technical scale**.
- Even a small but expressive prototype is valuable – let your imagination lead!

Tips for Participants

- Begin with a **clear emotion or story** – decide what you want your sound to express.
- Focus on **creativity and user experience**, not just complex models.
- Keep your code organized and your demo engaging.
- Let your **music and visuals** tell the story of your idea.
- Remember: originality, simplicity, and impact go a long way.