



**COMPUTER SOCIETY OF INDIA
COEP TECH STUDENT CHAPTER**



with



as our

Title Sponsor

presents

INSPIRON 4.0

A NATIONAL LEVEL HACKATHON

in association with



as our

Co-Title Sponsor

PROBLEM STATEMENTS

Problem Statement 1 - AI-Powered SQL Assistant

-by  e6data

Problem Statement:

Databases require expertise in schema design, SQL dialects, and query optimization. Many developers, analysts, and data engineers struggle with:

- **Designing a well-structured database schema** based on business needs.
- **Generating DDL (Data Definition Language) statements** to create tables.
- **Generating DML (Data Manipulation Language) statements** to insert, update, or delete records.
- **Translating natural language queries into SQL** while supporting different SQL dialects such as **Trino and Spark SQL**.
- **Incorporating user feedback** on query correctness and refining future query suggestions.

Your Challenge:

Build an **AI-powered SQL assistant** that can:

- **Suggest an optimal database schema** based on natural language descriptions for OLAP use cases.
- **Generate correct DDL and DML statements** for the requested schema.
- Provide **SQL completion assistance**.
- **Translate English text into SQL queries** in Trino or Spark SQL dialects, based on user selection.
- **Execute queries** using Trino or Spark SQL as execution engines.
- **Incorporate user feedback** to improve query generation and schema recommendations over time.

Technical Requirements:

- Dynamically generate SQL statements based on the chosen dialect (Trino/Spark SQL).
- Use a JDBC-compatible database or a mock execution environment for validation. Implement feedback learning to refine schema and query suggestions over time.
- Provide an **interactive UI** where users can:
 - Input schema requirements
 - Generate and edit SQL statements
 - Execute queries (optional)
- Provide feedback

Utilize **free and preferably open-source AI models** as required.

Bonus Features (For Extra Points!)

- Suggest a **storage-optimized and performance-efficient schema**.
- Leverage **specialized functions** based on the selected SQL dialect.

Evaluation Criteria

1. Accuracy

Generated SQL statements should be relevant and correct.

2. Functionality

- Proper use of **dialect-specific functions** (e.g., UNNEST for Trino vs. EXPLODE for Spark SQL).
- Correct usage of **PARTITION BY / CLUSTER BY** etc in table definitions.

3. SQL Translation Accuracy

- Proper use of SQL constructs, including **aggregations, joins, filters, limits, and ordering**
- Queries should be **both syntactically and semantically correct**.

4. Performance

- Generated DDL statements should be **optimized for performance**, incorporating best practices like partitioning
- Translated SQL queries should be **optimized for efficient execution**

5. User Interface

The UI should be **responsive, simple, and user-friendly**.

Problem Statement 2 - Metastore Viewer for Parquet, Iceberg, Delta & Hudi Tables on S3

-by  e6data

Background

E6data leverages **lakehouse formats** on object stores and requires an efficient way to explore metadata without relying on traditional metastore services. While many metastores allow registering **Parquet, Iceberg, Delta, and Hudi** tables, most do not support metadata exploration when only a **storage location** (such as an S3 bucket containing Iceberg tables) is provided.

Problem Statement

Develop a **web-based metastore viewer** that enables users to provide an **S3 (or Azure Object Store / MinIO) bucket path** containing **Parquet, Iceberg, Delta, or Hudi** tables and retrieve structured metadata, including (but not limited to):

- **Table Schema:** Column names, data types, partition keys
- **Table Properties:** Format-specific configurations (e.g., Iceberg's manifest files, Delta's transaction logs)
- **Partition Details:** Partition columns, pruning statistics
- **Versioning & Snapshots:** Iceberg's metadata.json , Delta's checkpoint files
- **Key Metrics:** File sizes, row counts, storage optimizations

Requirements

1. Backend

- Read **lakehouse file metadata** directly from S3 (or other object stores)
- Parse metadata files (e.g., Iceberg manifests, Delta logs) to extract table details
- Expose a RESTful API for the frontend

2. Frontend

- Accept S3 (or other object store) paths to fetch metadata
- Provide an interactive UI to browse schemas, partitions, and snapshots
- Display sample data from the table
- Visualize table evolution and statistics

3. Tech Stack

- Any library if used must be either open source or freeware for enterprise use.

Bonus Features

- **Trino Integration:** Execute queries directly on discovered tables
- **Metadata Comparison:** Track schema changes over time

Evaluation Criteria

- **Functionality :** Ability to parse and display metadata from object stores
- **Usability:** Intuitive and user-friendly UI
- **Performance:** Efficient handling of large and evolving datasets
- **Innovation:** Features such as snapshot comparison and change summaries are encouraged

Expected Outcome

A **metastore viewer** that allows users to explore **Iceberg, Delta, and other lakehouse table metadata stored in object stores**, making it easier to manage and analyze large-scale table formats without requiring a dedicated metastore service.

Problem Statement 3 - Job Portal Web Application

Background

Many job matching platforms are locked behind paywalls, creating financial barriers for both job seekers and recruiters. As a result, job seekers struggle to find opportunities, and recruiters face challenges in sourcing the right talent efficiently. This hackathon aims to address these issues by developing a free-to-use job matching platform that simplifies hiring and career growth for everyone.

Objective

The goal is to create an accessible and efficient job matching platform that eliminates financial barriers, ensuring free access to job seekers and recruiters while optimizing the hiring process.

Key Features

For Job Seekers:

- **Profile & Resume Management:** Create profiles and upload resumes easily.
- **Smart Job Search:** Find jobs using filters like location, experience, and industry.
- **Application Tracking:** Monitor job application statuses in real-time.

For Recruiters:

- **Free Job Posting:** Post and manage job listings at no cost.
- **Candidate Search:** Filter candidates by skills, experience, and qualifications.
- **Application Dashboard:** Review and manage applications efficiently.

Extra Features

For Job Seekers:

- Resume Builder: Create and customize resumes using an integrated tool.
- Interview Scheduling: Manage appointments with a built-in calendar.
- Job Alerts & Notifications: Get real-time updates on job postings and applications.

For Recruiters:

- Automated Communication: Send real-time updates to candidates via email.
- Collaborative Hiring Tools: Enable teamwork for managing listings and candidates.
- Analytics & Performance Metrics: Gain insights from job post engagement and candidate activity.

By removing financial constraints, this platform will bridge the gap between job seekers and recruiters, ensuring a smooth, cost-free hiring experience for all stakeholders.

Judging Criteria

Novelty / Innovation – The platform should introduce unique features or creative solutions that differentiate it from existing job matching platforms.

Usability (Ease of use: UI, UX) – The user interface and experience should be intuitive, ensuring seamless navigation for both job seekers and recruiters.

Implementation (Functionality) – The platform should be fully functional, with core features effectively implemented and working as intended.

Presentation Skills – The team should clearly articulate their ideas, demonstrating the platform's value, features, and impact effectively.

Adherence to Hackathon Timelines – The project should be completed within the given time frame, meeting all submission deadlines.

Problem Statement 4 - AI for Smart and Adaptive Systems

- by  **mira**

Background

Artificial Intelligence (AI) is transforming various industries by enabling smart, adaptive, and automated systems. From personalized learning experiences to AI-driven business intelligence, the demand for AI solutions that enhance efficiency, creativity, and decision-making continues to grow. However, many AI applications remain siloed, focusing on specific domains rather than offering holistic, multi-purpose solutions. This hackathon seeks to explore AI's potential across various domains, encouraging participants to develop smart and adaptive systems that address real-world challenges.

Objective

The goal is to build AI-powered solutions that can intelligently adapt to user needs, improve workflows, and enhance creative expression. By focusing on smart and adaptive systems, this track encourages participants to innovate across multiple domains, including education, business, software development, and creative content generation. Participants are free to choose one or more domains from the given focus areas.

Key Focus Areas

- Personalized Learning – AI-driven tutors that adapt to student progress, automated exam preparation assistants, and accessibility tools like text-to-audio summarization.
- AI for Developers – AI-powered code completion, automated bug detection and fixing, and intelligent documentation generation for improved software development workflows.
- Business Intelligence – AI-driven financial insights, predictive analytics for decision-making, and smart automation tools for enterprises.
- Creative AI – AI-generated music, automated video editing assistants, AI-powered storytelling or meme generation.

By integrating AI across these domains, participants will have the opportunity to create versatile, impactful solutions that showcase the true potential of smart and adaptive AI systems. The challenge is to build innovative, functional, and user-friendly AI applications that address real-world problems while demonstrating technological excellence.

Evaluation Criteria

- **Novelty / Innovation** – The AI solution should introduce unique features or creative approaches that differentiate it from existing technologies.
- **Usability (Ease of use: UI, UX)** – The user interface and experience should be intuitive, ensuring seamless interaction and accessibility.
- **Scalability** – The AI solutions should be designed to efficiently handle growing datasets, user interactions, and evolving demands without compromising performance.
- **Implementation (Functionality)** – The solution should be fully functional, with core features effectively developed and working as intended.
- **Presentation Skills** – The team should clearly communicate their ideas, demonstrating the solution's value, features, and impact effectively.
- **Adherence to Hackathon Timelines** – The project should be completed within the given timeframe, meeting all submission deadlines.