Project Progress Report — Team 04 AI Copilot for Databases (Text-/Voice-to-SQL with Safe Execution & Explanations)

Team Members: Kartavya (Architecture & Ranking), Kanav (UX & User Studies), Saarthak (PM & Documentation)

1. Problem Statement

Relational databases remain inaccessible to many non-technical users. Writing SQL requires remembering schema details and syntax, while existing AI tools often generate incorrect or unsafe queries. These issues slow decision-making and reduce trust in AI systems. Our project addresses the problem of making databases conversationally accessible through a safe, verifiable, and explainable text/voice interface.

2. Project Goal

By the end of the semester, our goal is to deliver a working prototype that can:

- Connect securely to a user-specified SQL database (starting with PostgreSQL).
- Translate a text or voice query into SQL safely and accurately.
- Explain the query logic and display results in an intuitive interface.
- Demonstrate at least 80–85% correctness on benchmark queries, and zero unsafe query generation.

3. Current Status

Although the full NLP-to-SQL system is not yet active, we have established the **technical foundation** for it:

- Database Connection Setup: Implemented connection via URL and manual credential input. The system can now connect to a live PostgreSQL database and fetch table and column metadata securely.
- Schema Crawler (Partial): Extracts and displays database schema for the user in the app's setup phase, forming the base for context-aware query planning later.
- NLP Pipeline Initialization (Partial): Laid the foundation for the text-processing and prompt-handling modules (using an API-based LLM backend). The pipeline can parse and tokenize queries but is not yet linked to SQL generation.
- Frontend Integration: Basic interface built using a simple form for question input and database linking. Voice input and result display components have placeholders ready.
- Version Control & Environment Setup: Project hosted on GitHub with modular structure, enabling independent work on UI, planner, and evaluation modules.

4. Evaluation Plan

Our evaluation will focus on functionality and user experience once the system is functional.

Dimension	Metric	Target	Method
Connection Success	% of valid database connections	≥ 95%	10+ test databases with varied schemas
Translation Accuracy	Correct SQL or correct result set	≥ 75%	50 benchmark NL–SQL pairs (Spider subset)
Safety	% of unsafe (write/DDL) queries	<10%	Static checks before execution

5. Partial Results

At this stage, the project can successfully connect to local and hosted PostgreSQL databases and display schema structure. The NLP parser loads correctly, and we have laid the foundation for it. We need to do more testing to assess how well it works.

6. Plan Until the End

(Oct 14-20)	Integrate NLP parser with query generation module; basic mapping from text to SQL templates.
(Oct 21–27)	Implement safety filters and schema-aware validation. Begin limited SQL translation tests.
(Oct 28–Nov 3)	Add voice input; refine interface; run first small-scale user demo.
(Nov 4–10)	Conduct evaluation on test dataset; finalize report and demo.

7. Team Member Contributions

Kartavya: Database connector, schema crawler, ranking and safety gate logic.

Kanav: UI / UX design, usability study setup, prototype testing, metrics dashboard.

Saarthak: Project management, write-ups, version control, evaluation dataset creation.