Milestone 3
Software engineering

Zeina Shebl 202200080
Malak Hesham 202200011
Malak Ehab 202200069
Mariam Sherif 202200825

Initial prompt:

Write a well-structured, high-quality prompt that instructs an AI to design and implement a complete, visually appealing web-based Graphical User Interface (GUI) for a Database Management System (DBMS) connected to a Supabase PostgreSQL database for a fitness management system. The prompt you write must clearly define the AI's role, provide sufficient system and database context, specify frontend and backend technologies, outline required CRUD functionality, include expectations for UI/UX design quality, and mention enhancing existing SQL code for clarity, performance, and data integrity without altering the core schema or business logic. The resulting prompt should be concise but detailed, enforce clear constraints, avoid ambiguity, and guide the AI toward producing a complete frontend implementation, Supabase integration, enhanced SQL queries, and example API calls.

Prompt 1:

Assume the role of an experienced full-stack web developer and database engineer tasked with building a complete web-based Graphical User Interface (GUI) for a Database Management System (DBMS) connected to an existing Supabase PostgreSQL database for a fitness management system. The database already contains the entities Admin, Coach, Client, Membership, Nutrition_Plan, and Workout_Plan, with all tables, relationships, and SQL code fully implemented and reused without modification. Your responsibility is to design and implement a clean, intuitive, and user-friendly interface using HTML, CSS, and JavaScript (or React if preferred), integrating directly with the Supabase backend through its official JavaScript client. The GUI must support full CRUD functionality, including creating new records, displaying data in structured tables, updating existing records such as client assignments to coaches or plans, and safely deleting records while respecting foreign key constraints. The system should also provide basic data views, such as listing all clients with their coaches, viewing membership types, and displaying nutrition and workout plans. The final deliverable must include the complete frontend implementation, Supabase connection setup, and

Milestone 3
Software engineering

Zeina Shebl 202200080
Malak Hesham 202200011
Malak Ehab 202200069
Mariam Sherif 202200825
example API calls for insert, update, delete, and select operations, with all requirements implemented exactly as specified and without introducing assumptions or changes to the existing database structure.

Prompt 2: (enhanced display)

Assume the role of a senior full-stack web developer with strong UI/UX expertise and database engineering experience. Your task is to build a complete, visually appealing, and responsive web-based Graphical User Interface (GUI) for a Database Management System (DBMS) connected to an existing Supabase PostgreSQL database used for a fitness management system. The database already includes the entities Admin, Coach, Client, Membership, Nutrition_Plan, and Workout_Plan, with all tables, relationships, and SQL code fully implemented and reused without modification. Design and implement the frontend using HTML, CSS, and JavaScript (or React if preferred), ensuring a modern, professional look with consistent colors, typography, spacing, and layout. The interface should include a clear navigation structure, well-styled forms, responsive tables, and intuitive action buttons, with visual feedback such as hover states, validation messages, and confirmation alerts.

The GUI must connect directly to the Supabase backend using the official JavaScript client and support full CRUD operations, including creating new records, viewing and filtering data in tables, updating existing records such as client assignments to coaches or plans, and safely deleting records while respecting foreign key constraints. It should also provide common data views such as listing all clients with their coaches, displaying membership types, and showing nutrition and workout plans. The final deliverable must include the complete frontend implementation, Supabase connection setup, and example API calls for insert, update, delete, and select operations, with all requirements implemented exactly as specified and without introducing assumptions or changes to the existing database structure.

Milestone 3
Software engineering

Zeina Shebl 202200080
Malak Hesham 202200011
Malak Ehab 202200069
Mariam Sherif 202200825

Prompt 3: (enhanced display and fixed sql code)

Assume the role of a senior full-stack web developer with strong UI/UX expertise and advanced database engineering skills. Your task is to build a complete, visually appealing, and responsive web-based Graphical User Interface (GUI) for a Database Management System (DBMS) connected to an existing Supabase PostgreSQL database used for a fitness management system. The database includes the entities Admin, Coach, Client, Membership, Nutrition_Plan, and Workout_Plan, with tables and relationships already defined; however, the SQL code should be reviewed and enhanced where necessary to improve clarity, consistency, performance, and data integrity (e.g., optimized queries, clear naming conventions, proper constraints, indexes, and safe cascading rules) without altering the intended schema or business logic. Design and implement the frontend using HTML, CSS, and JavaScript (or React if preferred), ensuring a modern, professional visual design with consistent colors, typography, spacing, and layout. The interface should feature intuitive navigation, well-styled and validated forms, responsive data tables, and clear action buttons with visual feedback such as hover states, confirmation dialogs, and error handling.

 The GUI must connect directly to the Supabase backend using the official JavaScript client and support full CRUD operations, including creating new records, viewing and filtering data, updating existing records such as client assignments to coaches or plans, and safely deleting records while respecting foreign key constraints. It should also provide essential data views, such as listing all clients with their coaches, displaying membership types, and showing nutrition and workout plans. The final deliverable must include the complete frontend implementation, Supabase connection setup, enhanced and well-structured SQL queries, and example API calls for insert, update, delete, and select operations, with all requirements implemented precisely and without introducing unintended changes to the existing database design.

Milestone 3
Software engineering

Zeina Shebl 202200080
Malak Hesham 202200011
Malak Ehab 202200069
Mariam Sherif 202200825

Techniques used:

Role-Based Instruction:

The prompt starts by assigning a clear role:

- Assume the role of a senior full-stack web developer

Why it works: This sets expectations for expertise, decision-making level, and output quality. It guides the response toward professional-grade solutions rather than basic examples.

Context based:

Detailed background information is provided:

- Fitness management system
- Existing Supabase PostgreSQL database
- Defined entities and relationships

Why it works: Providing context reduces ambiguity and ensures the output matches the real system instead of generic assumptions.

Constraint Specification:

The prompt clearly states limitations:

- SQL schema already exists
- Business logic must not change
- Enhancements allowed only for clarity, performance, and integrity

Why it works: Constraints prevent unwanted changes and keep the solution aligned with project requirements.

Milestone 3
Software engineering

Zeina Shebl 202200080
Malak Hesham 202200011
Malak Ehab 202200069
Mariam Sherif 202200825

Task Decomposition:

The request is broken into logical responsibilities:

- UI/UX design
- Frontend implementation
- Database interaction
- SQL enhancement
- CRUD operations

Why it works: This ensures all required components are addressed and nothing important is skipped.

Quality & Output Expectations:

- The prompt specifies desired qualities:
- Visually appealing and responsive UI
- Optimized and well-structured SQL
- Safe database operations
- Complete deliverables

Why it works: Defining quality standards improves the usefulness and completeness of the result.