StudyGen 360

StudyGen 360 AI
From Plain-Language Prompt to a Working Study Report



AI Study AI

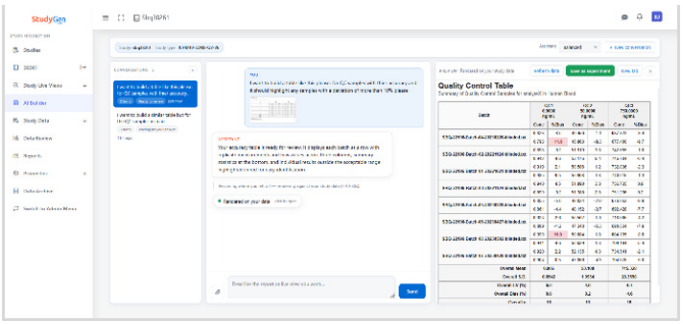
- Turns plain language into a configuration that builds ad hoc tables and graphs behind your reports
- Builds experiment layouts around a study's existing structure
- Deterministic, standard configuration output
- Two-layer Envelope-Core architecture underneath
- Automatic hand-over to the AI covers data structure only, never results

From Plain-Language Prompt to What Your Lab Actually Needs

A scientist describes what they need in plain language: a new experiment layout, a report structure, a set of graphs, or summary tables. **StudyAI**, the chatbot built into StudyGen 360, turns that description into a configuration, which StudyGen 360 then runs against the study's real data to produce tables, images, and reports, already populated with the study's own results. No developer, no queue, the answer lands straight in the scientist's hands, in the same conversation in real-time. And most importantly, the platform passes only structural details to the AI, such as column names and formats, and the study's results stay inside StudyGen 360. The one other input is whatever a user types or pastes into the chat.

Checked Against a Seed Data Set Before You Might See It

Before any result reaches a user, StudyAI runs the generated experiment against an anonymized seed data set to verify its correctness. What emerges is a standard, deterministic configuration template compatible with existing applications such as *Expression Studio*, allowing a lab to validate it using the same processes it already trusts, rather than treating it as a black box.



Human Review at AI Solutions Is Non-Negotiable. We Just Made It Take Seconds.

In a GLP- and ICH M10-regulated environment, the FDA doesn't leave room for interpretation: AI-generated data

must be reviewed by a human before it counts. That requirement isn't going anywhere — no architecture, however advanced, can validate itself out of that rule. So, we didn't try to remove the review. We redesigned everything that happens before it. The starting point changes: because the two-layer architecture has already validated the configuration against seed data before a reviewer ever sees it, that review moves faster, confirming a result rather than building one from scratch. Source data stays visible next to every AI-generated table or image, so a reviewer can check a claim in seconds instead of tracing it back by hand, and that decision becomes part of the study's audit trail, like any other review.

Two Layers Under the Hood

Under the hood, StudyAI runs on a two-layer architecture. The blueprint is a deterministic envelope around the generative core, handling input and output validation and maintaining an immutable audit trail, all validated the way any conventional system would be. Inside it, the model, its prompts, and its retrieval sources are versioned configuration rather than anything hardcoded into the application. The AI itself sits behind three explicit contracts: a typed input, a typed output, and a defined error path, so it behaves as a contained component rather than something that can quietly affect the rest of the system. If it fails, the workflow degrades to a manual path with an appropriate alert, rather than presenting an AI result as though it were an ordinary deterministic one.

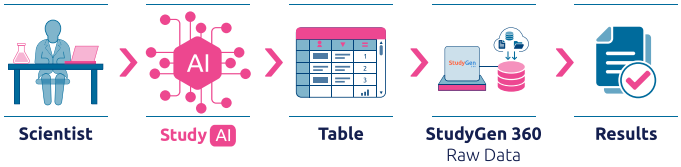
Every System Update Starts With a Risk Assessment

Updates to the AI itself, its underlying model, its core instructions, and the documentation references to build a configuration go through a formal review before any deployment. The change is classified using a risk assessment, its potential impact on affected outputs is evaluated, it's re-qualified with a regression check scaled to that risk level, a named release decision is recorded, and the entire sequence is captured in the audit trail. The risk assessment scores potential damage against the likelihood of catching an error before it reaches a result, and higher-damage outputs are held to materially stricter re-qualification regardless of how minor a given change appears. A lab user's own prompts, what they type to build or refine a report in the moment, are a separate matter: those are reviewed the same way as any other AI output, through the human checkpoint already in place.

Built to Share Structure, Not Results

By design, the platform hands the AI only an abstract model of the data structure, such as column names, formats and filter keywords, plus the experiment specification. Measurement results are calculated inside StudyGen 360. This architecture is described in an in-progress methodology paper developed alongside up to data's implementation.

Scientist Study Data Processing Workflow



From question to result in four steps: The scientist works with StudyAI, which structures the input into a defined table format. StudyGen 360 then combines this structured data with the raw study data to deliver traceable results.

At a Glance

- Tables, graphs & full reports from plain language
- Experiment layouts built directly from study data
- Validated against anonymized seed data
- Deterministic, standard configuration template
- Compatible with existing apps *(e.g. Expression Studio)*
- Two-layer Envelope-Core architecture
- Human checkpoint before existing GLP/M10 review
- Platform shares data structure, not results

Ready to Experience the Power and Efficiency of StudyGen 360?