STRUCTURE

Inhaltsverzeichnis

docs/STRUCTURE.md

← Back to README

Root structure (high level)

- app/
  Next.js App Router
  - Public study flow UI
  - Management UI
  - API routes (auth, study flow, transcript export, etc.)
- components/
  Shared UI components, including chat/thread UI components derived from LangChain Agent Chat UI patterns
- app/modules/ My own components
- app/modules/langgraph/
  LangGraphJS agents, prompts, tools, persistence helpers, and local dev/test scripts
- data/
  Dataset CSV files (raw/ and processed/) + dataset manifest
- db/
  Dataset DB init SQL scripts (schema + COPY import)
- prisma/
  Prisma configuration and commands for the App DB

Public study flow routes (app/(public))

Entry and routing:

- /study
  Access code entry
- /study/[accessCode]/pre
  Pre-survey

- /study/[accessCode]/task/[taskNumber]
  Task chat (1..3)
- /study/[accessCode]/task/[taskNumber]/post
  Post-task survey
- /study/[accessCode]/final
  Final survey
- /study/[accessCode]/done
  Done screen / completion

Key UI file:

- app/(public)/page.tsx
  Public landing page with dataset + study context

Management routes (app/(management))

Protected admin area:

- Participants overview + details (incl. transcript export)
- Survey builder and analytics
- Dashboard overview (study progress, metrics, etc.)

LangGraph agents (app/modules/langgraph/agents)

- dataAwareLLMSystem/
  Main study agent
  - SQL querying tool for dataset DB
  - schema introspection tool
  - persistence/logging for “data insights” (thread-level DQ logs)
  - prompt definitions for the study assistant
- tavilyAgent/
  Dev-only agent used during development (kept in repo, not required for study execution)

Dataset DB (db/ + data/)

DB init scripts:

- db/esg/init/00_schema.sql
  Creates schema `esg` and tables
- db/esg/init/10_load.sql
  Imports CSV via COPY into the tables

CSV locations:

- data/raw/companies.csv
- data/raw/indicator_metadata.csv
- data/processed/esg_indicators_postprocessed.csv

Dataset manifest:

- data/dataset-manifest.json
  Maps dataset files to SQL tables, used by the system to render dataset/source metadata in the UI.

App database (Prisma) overview

Prisma schema (App DB) is designed for:

- reproducible study configuration (tasks/surveys stored in DB)
- analytics-friendly logging (timestamps/counters)
- append-only chat transcripts (user + assistant + tool logs)

Main areas in the schema:

1) Administration
   - User (management login, optional 2FA fields)
2) Study / participants
   - Study
   - Participant (assigned variant, current step, progress timestamps)
   - ParticipantAccessLog (re-entries, audit)
3) Tasks
   - TaskDefinition (task prompt stored for reproducibility)
   - TaskSession (per task run; metrics like message counts, side panel usage)
4) Chat transcript (append-only)
   - ChatThread (each restart creates a new thread)
   - ChatMessage (ordered by sequence; user + assistant + tool messages)
   - ChatToolCall (minimal tool-call logs)
5) Surveys
   - SurveyTemplate, SurveyQuestion, SurveyOption (definition)
   - SurveyInstance, SurveyAnswer, SurveyAnswerOption (responses)
6) Dataset catalog metadata (for Data Insights UI)
   - Dataset, DatasetTable (maps dataset files to SQL tables)
7) Thread-level data insights logging
   - ThreadDataQualityLog (stores latest indicators + used tables per LangGraph thread_id)

This design supports later quantitative analysis (durations, counts, feature usage) and qualitative inspection (transcripts).

Where to start reading code

- Main public landing context: app/(public)/page.tsx
- Study chat flow: app/modules/publicStudy/components/TaskChatFlowClient.tsx
- Agent implementation: app/modules/langgraph/agents/dataAwareLLMSystem/graph.ts
- SQL tool + guards: app/modules/langgraph/agents/dataAwareLLMSystem/tools/
- Data Insights UI: components/thread/data-insights/

← Back to README