Attri AI-Engineer Intern Take-Home Assignment

Thanks for spending time on this! We're a small team and we genuinely value your effort. This exercise is about seeing how you think, frame problems, and find solutions.

1. Scenario & Goal

Build a **natural-language bike-share analytics assistant**.
Example user question:

> "How many kilometres did women ride on rainy days last month?"

2. Resources Provided

Item	Details

PostgreSQL 17 instance	PGHOST=agentify-assessment.postgres.database.azure.com
	PGUSER=attriassessment
	PGPORT=5432
	PGDATABASE=bike-share-assessment
	PGPASSWORD=(.aG0X>322Uk

Nothing else	Discover the schema yourself at runtime.

3. Functional Requirements

#	Requirement	Notes
F-1	Simple chat-style UI (web or desktop)	**TypeScript, NodeJS, and/or Python**.
F-2	Translate questions into **parameterised SQL** (no raw string concatenation).	Prevents SQL injection.
F-3	**Semantic column discovery** – user words rarely match column names.	No hard-coded synonym lists.

F-4	Support single-table filters, joins, aggregations, date math (“last month”), group-by.	
F-5	Graceful error handling (unknown intent, empty result set, etc.).	
F-6	Expose a public HTTP endpoint `/query` that accepts `{ "question": "<user-text>" }` and returns: `{ "sql": "<final-query>", "result": <rows | scalar>, "error": <null | "message"> }`	
F-7	Unit tests covering your SQL-generation logic **plus** the three public sample questions (§7).	Any test runner.
F-8	`README` (max ≈ 2 pages) explaining architecture, decision doc and semantic-mapping method	

4. Technical Constraints

- LLM calls are **allowed but insufficient**:
  - Introspect `information_schema.columns` dynamically.
  - Score candidate tables/columns vs user text.
  - Choose mappings deterministically.
- No hard-coded English→column synonyms.
- Keep secrets out of source control (use env vars).
- Must run on Linux; Docker appreciated.

5. Timeline & Submission

- **3 business days** from receipt of credentials.
- Submit a Git repo or zipped directory containing:

/src
/tests

```
README.md
LICENSE
```

6. We’ve Solved This Before 👍

This mirrors challenges we already tackle internally; it exists to see how you reason under uncertainty. We appreciate the time you’ll invest.

7. Public Acceptance Tests (must pass all three)

Test #	Natural-language request	Expected answer*
T-1	“What was the **average ride time** for journeys that **started at Congress Avenue** in June 2025?”	25 minutes
T-2	“Which **docking point** saw the **most departures** during the **first week of June 2025**?”	Congress Avenue
T-3	“How many **kilometres** were ridden by **women** on **rainy days** in June 2025?”	6.8 km

*Numeric/string values must match; wording can vary.

Heads-up: Our grading harness also runs **unpublished tests** probing new synonyms, derived attributes, and security exploits. We suggest generating extra synthetic questions with AI tools and adding regression tests so you’re ready for surprises.

8. Evaluation Rubric (100 pts)

Category	Points
Public tests (T-1 – T-3)	20
Hidden tests (edge-cases & security)	20
Semantic mapping quality	15
Code structure & clarity	15

Security / SQL-injection safety	10
UI/UX polish	**15**
Documentation & unit tests	5

Good luck—show us how you think!