CURRICULUM VITAE · DATA ENGINEERING

Tahmid Jawad Shafi

Dhaka (UTC+6) Open to internships and full-time roles from January 2027
tahmidshafi1050ti@gmail.com

EDUCATION

BSc CSE, AIUB — AIUB (Dhaka) June 2027

PROJECTS

Bari-Link (https://tahmidshafi.me/work/bari-link) featured 2026-11 – 2026-11

Designed and shipped pipeline completed

Python · Splink · DuckDB-Wasm · React · TypeScript · Vite · Pytest · Playwright

Deduplicates NGO beneficiary registries that share no identifier, using kinship constraints to cut costly false merges.

- Built a synthetic population engine and noise model, romanised-name normalisation, and a Splink Fellegi-Sunter model with kinship cannot-link constraints (Pipeline K).
- Added multi-channel household inference on top of the resolved person clusters.
- Committed hypotheses before their results and confirmed the champion pipeline on five fresh holdout seeds.
- Shipped a static DuckDB-Wasm demo of the results.
- Operational cost at a 5:1 false-merge cost ratio, Pipeline K vs the Splink reference (holdout seeds 5–9, lower is better): **812.2** reference 888.2; -8.56%; better on 5/5 seeds, one-sided sign test p = 0.031 (source: README.md:48–50 @ c518135)
- Cluster-level precision, Pipeline K (holdout seeds 5–9): **0.9656** ± 0.0106; Splink reference 0.9457 ± 0.0127 (source: README.md:47–48 @ c518135)

Repository (https://github.com/TahmidShafi/bari-link)

Demo (https://tahmidshafi.github.io/bari-link/)

Curbstone (https://tahmidshafi.me/work/curbstone) featured 2026-11 – 2026-11

Designed and shipped quality completed

Python · SQL · DuckDB-Wasm · Parquet · Pytest

Ranks survey enumerators by risk of fabricating interviews that range and skip checks let through.

- Designed peer-comparison features (interview duration, daily throughput, near-duplicate answers, answer dispersion, GPS spread, age heaping, non-response) and a frozen risk-scoring engine.
- Built a fabrication simulator with five fraud modes plus one mode held out from development, so detection performance is measured instead of asserted.
- Ran the same 12 SQL rule checks in Python and DuckDB-Wasm with a cross-engine parity check, which caught a timezone bug.
- Precision@4 gain over the rule-count baseline, 95% CI (72 configurations × 20 rounds): **+0.131** interval +0.119 to +0.142 (source: README.md:19–21 @ 363c7b4)
- Average-precision gain over the rule-count baseline, 95% CI: **+0.195** interval +0.178 to +0.211 (source: README.md:19, 21 @ 363c7b4)
- Held-out fraud mode (partial_section) detected, risk ranking vs rule checks: **11/15** rules: 1/15 (source: README.md:37 @ 363c7b4)

Repository (https://github.com/TahmidShafi/curbstone)

Demo (https://tahmidshafi.github.io/curbstone/)

Nittoo (https://tahmidshafi.me/work/nittoo) featured 2026-09 – 2026-11

Designed and shipped analytics completed

TypeScript · React · Supabase · PostgreSQL · Python · Pydantic · DuckDB · dbt · Vitest · Playwright

Turns everyday purchases into container lifespans, true daily costs and calibrated run-out windows.

- Built the React + Supabase app: container-level tracking, run-out forecasting and restock reminders, with Row-Level Security tenant isolation verified in CI.
- Built the Forecast Lab: a Pydantic + JSON Schema export contract, a seeded household simulator, idempotent DuckDB ingestion with a rejects quarantine table, and a dbt warehouse.
- Backtested forecasts with rolling-origin evaluation and exported calibrated interval quantiles and parity fixtures back into the app.
- Overall holdout forecast error (MAE), size-normalised rate forecast shipped in the app vs the app's original average-duration baseline, synthetic data: **4.53 days** baseline 17.42 days (source: README.md:89–93 @ b0fb0fb · rate_mean (rule_only) vs baseline (none))
- Forecast error (MAE) after pack-size changes, size-normalised rate forecast shipped in the app vs the app's original average-duration baseline, synthetic data: **4.88 days** baseline 21.59 days (source: README.md:89–93 @ b0fb0fb · rate_mean (rule_only) vs baseline (none))
- P10–P90 run-out window coverage on holdout, size-normalised rate forecast shipped in the app, synthetic data: **83.7%** nominal 80%; mean width 12.8 days (source: README.md:95 @ b0fb0fb · rate_mean (rule_only))

Repository (https://github.com/TahmidShafi/nittoo) Demo (https://nittoo.vercel.app)

ProtoMacro (https://tahmidshafi.me/work/protomacro) featured 2026-09 – 2026-11

Designed and shipped app completed

JavaScript · Python · DuckDB · SQL · Pandas · Vite · Chart.js · Vitest · Pytest · Playwright

Grounds a nutrition tracker's 567 foods in Bangladeshi and USDA reference tables instead of unsourced hand-entered values.

- Built a tested DuckDB pipeline that extracts, SHA-256-verifies, normalises and validates FCTB 2013 and USDA FoodData Central data into a catalog with provenance on every row.
- Audited the legacy hand-entered values against exact primary-source matches and migrated saved v2 IDs to v3 without altering logged history.
- Shipped an offline-first PWA (Vanilla JS + Workbox) with Vitest, pytest and Playwright checks in CI.
- Foods in the sourced catalog, each with row-level provenance: **567** 381 FCTB 2013 · 69 USDA FDC SR Legacy · 117 labelled estimates (source: README.md:11, 63 @ e635153)
- Calorie error found in the legacy hand-entered data (n = 74): **2.5% / 34.9%** median / 90th percentile, absolute (source: README.md:21 @ e635153)

Repository (https://github.com/TahmidShafi/ProtoMacro)

Demo (https://tahmidshafi.github.io/ProtoMacro/)

DGHS CleanPulse (https://tahmidshafi.me/work/dghs-cleanpulse) 2026-10 – present

Designed and built pipeline in-progress

Python · Pandas · Pandera · DuckDB · Parquet · FastAPI · Next.js · Pytest

Validates Bangladesh dengue and district climate data through a Medallion-architecture pipeline with no silent data drops.

- Built a Medallion pipeline whose Bronze layer stores immutable raw responses, each paired with a SHA-256 provenance sidecar.
- Built a Pandera validation firewall that records anomalies instead of dropping rows silently, tested against a synthetic fault-injection fixture.
- Built a FastAPI + DuckDB analytics API and a Next.js console.

Repository (https://github.com/TahmidShafi/dghs-cleanpulse)

SKILLS

Languages	Python, SQL
Data & Storage	DuckDB, DuckDB-Wasm, Parquet, PostgreSQL
Transform	dbt, Pandas, Splink
Validation	Pydantic, Pandera
Testing	Pytest

LANGUAGES

Bangla

English

LINKS

GitHub	https://github.com/TahmidShafi (https://github.com/TahmidShafi)
LinkedIn	https://www.linkedin.com/in/tahmid-jawad-shafi-70a271268/ (https://www.linkedin.com/in/tahmid-jawad-shafi-70a271268/)
Email	tahmidshafi1050ti@gmail.com
Portfolio	https://tahmidshafi.me (https://tahmidshafi.me)