Full-Stack Take-Home Assignment (48 Hours)

Issued: 16 Jan 2026	Submission: GitHub repo link + 5-10 lines summary

Goal: Upload a CSV, view it as a table, select a column, compute statistics, and plot a histogram.

Dataset: One sample CSV (~1000 rows) will be provided. Your solution must work for this file.

Functional Requirements (UI)

- Browse / Upload CSV button.
- Render uploaded data as a sheet-like table (main area).
- Right-side panel with: column selector, statistic buttons, and histogram action.

Computation Rules (Mandatory)

- Numeric stats (min/max/mean/median) only for numeric columns; otherwise disable or show Not applicable.
- Mode: categorical = most frequent value; numeric = most frequent exact value. If ties occur, document tie-handling in README.
- Missing/invalid values: ignore in numeric calculations; show missing count in the side panel (recommended).
- Histogram: numeric columns only; 30 bins by default; show a clear message if no valid numeric data.

Backend Requirement (Mandatory)

Implement server-side CSV parsing and server-side statistics + histogram data generation. The frontend renders results from the API.

Minimum API Contract (Required)

- POST /api/upload -> dataset_id, schema summary (columns + types)
- GET /api/dataset/{id}/table -> rows for display (ok to return all ~1000)
- GET /api/dataset/{id}/column/{col}/stats -> stats JSON + missing count
- GET /api/dataset/{id}/column/{col}/hist -> histogram bins + counts

Constraints and Deliverables

- Time limit: 48 hours from receipt.
- Run locally; clear setup and run instructions must be provided in the README (Docker preferred, not required).

- Safety: accept CSV only; handle empty/malformed files with readable errors.
- Deliverables: GitHub repo (frontend + backend) + README (setup, run, endpoints, assumptions, limitations).
- Submission: Candidates must follow the submission guidelines provided in the README of the official assignment repository: https://github.com/career-dromolys/DromoLys_FullStack_Assignment

Optional Additions (Not Mandatory)

- Optional extras may be used for further evaluation (no penalty if core works).
- Examples: table sorting/search, configurable bins, better type inference, tests/CI, export stats.