ASSIGNMENT

Advanced Data Grid (Virtualized, Editable, Accessible)

Objective

Build a **production-grade data grid component** capable of handling very large datasets with strict guarantees around performance, accessibility, and correctness. This is a **core design-system primitive**, not an application feature.

Mandatory Tech Stack

This assignment **must strictly follow** the Global Constraints document:

- React 18+
- TypeScript (strict mode enabled)
- Tailwind CSS (utility-first, tokenized)
- Storybook + Chromatic (public)
- Vite or Next.js (App Router only)

Scope

- Support **50,000+ rows** using **manual row and column virtualization**
- Sticky headers and **pinned columns**
- Multi-column sorting
- Column resizing, reordering, and visibility toggles
- In-cell editing with async validation (mocked)
- Full keyboard-first interaction model
- Screen reader parity with visual UX

Explicitly Forbidden (Reinforced)

- No `react-table`, `tanstack/table`, `react-virtualized`, `react-window`
- No component or headless UI libraries
- No prebuilt grid, table, or virtualization utilities
- No copied implementations from blogs, GitHub, or demos

Violations result in **immediate disqualification**.

Tasks

1. Design a column/row schema supporting renderers, editors, validators, and metadata.
2. Implement **custom virtualization logic** for rows and columns.
3. Ensure pinned columns remain aligned during scroll, resize, and reorder.
4. Implement multi-sort with deterministic ordering.
5. Add in-cell editing with optimistic UI and rollback on simulated failure.
6. Define a complete keyboard contract and ARIA grid semantics.
7. Create comprehensive Storybook stories covering scale, failure, and accessibility modes.

Strict Requirements

- Sustains **60 FPS scrolling** on 50k rows (measured and reported)
- Zero layout shift during interactions
- Full keyboard operability with no focus traps
- Screen readers announce position, state, and errors correctly
- Undo support for column actions and edits
- Storybook must be published publicly

Deliverables

- DataGrid component with documented API
- Performance report (FPS, memory, interaction latency)
- Accessibility report (manual + axe)
- Public Storybook and Chromatic links