Workflow / Automation Builder — Frontend-Only (VueJS)

Build a **node-based workflow builder** as a **pure frontend app** using VueJS. Focus on canvas UX, dynamic configuration forms, undo/redo state management, local persistence, and a visual **Run Preview (simulation)**.

🎯 Goals

- **Canvas & Graph UX:** Drag/drop from a palette, connect nodes, edit edges, pan/zoom, fit-view, and mini-map.
- **Dynamic Forms:** Schema-driven config panels with validation and helpful error messages.
- **State & History:** Immutable updates, batched undo/redo, predictable and modular state management.
- **Persistence (Local):** Save/load workflows (graph + viewport) using LocalStorage/IndexedDB.
- **Run Preview:** Simulate workflow execution visually with play, pause, and step functionality.

🧩 Tech Stack (Recommended)

- **Framework:** VueJS 3 + TypeScript
- **UI Library:** Any Components or Library
- **Graph Library:** VueFlow or custom canvas implementation
- **State Management:** Pinia or Vuex with Immer for immutability

- **Validation:** Vee-Validate or Zod
- **Styling:** TailwindCSS

✅ Feature Checklist

- **Palette → Canvas:** Add, duplicate, delete nodes; snap-to-grid; multi-select; keyboard shortcuts.
- **Edges:** Connect/rewire via handles; label edges (true/false for conditions); delete connections.
- **Config Panel:** Opens on node select; fields generated from schema; live validation; Save disabled until valid.
- **Node Types (min):** Trigger: Manual/Webhook | Actions: HTTP, Email/SMS | Logic: Condition, Transform.
- **Graph Rules:** Acyclic (except optional Loop); typed ports; required labels for conditional branches.
- **Undo/Redo:** Keyboard support (Ctrl/Cmd+Z, Shift+Ctrl/Cmd+Z); batch multi-drag actions; safe time-travel.
- **Persistence:** Autosave (debounced) + explicit Save/Load; restore viewport and zoom.
- **Run Preview:** Executes in topological order; highlights active nodes; shows logs/status per step (success/error/skipped); includes play/pause/step controls.
- **A11y & Performance (Bonus):** Focus rings, roles/aria, reduced motion support; handle ~200 nodes/400 edges smoothly. (**Bonus point**)

🧪 Testing (Minimum)

- Canvas flows: create → connect → configure → save → reload → preview.

- Validation: ensure invalid inputs disable Save.
- Undo/Redo: verify state rollback.
- Persistence: confirm saved workflow reloads correctly.

📦 Deliverables

- Runnable app (`npm run dev`)
- **README (short):** architecture overview, state shape, how to add a node type, undo/redo logic.
- **2 Sample Workflows (JSON)** for demonstration.
- Passing **test script** with sample cases.
- **Deployment link of the working code**
- **Please share the recorder loom link of working example (Demo)**

💡 Example Use Cases

1️⃣New Lead Welcome & Follow-up

Webhook trigger (new lead) → Send Email (welcome) → Delay (1 hour) → HTTP Request (add lead to CRM) → Condition (if response.status == 200) → True branch: Send SMS (success message) → False branch: Send SMS (error alert to support).

2️⃣Abandoned Cart Recovery & Inventory Hold

Webhook trigger (cart event) → Condition (item in stock) → Email #1 (reminder) → Delay (2h) → Condition (still not purchased) → SMS nudge → HTTP create temp discount & hold inventory → Email #2 with code → Delay (24h) → Condition (purchased?) → True: HTTP release hold →

Email “Thanks for your order” → Log analytics; False: HTTP cancel hold → Email “Last-chance offer” → Log analytics.