Octodrop / Odoo-Style UI – Context Sync File

This document is meant to be pasted into a **new ChatGPT conversation** to fully restore context about the current state of the Octodrop product, manifest system, and UI direction.

What is being built

We are building **Octodrop**, an Odoo-like, manifest-driven application platform.

Key principles: - Manifests describe **intent**, not UI implementation details - UI is rendered by a strict runtime (FastAPI + React + DaisyUI) - No free-form layouts; everything is structured, validated, and AI-safe - Goal: allow AI to reliably generate *good* ERP-style apps (not garbage)

Core UX Goal (Very Important)

The UI must behave like Odoo.

That means:

List screens

- A list view is its own screen
- Has an Odoo-style header:
- New button
- Search
- Filters
- Bulk actions
- Clicking a row navigates to a **form page** (not side-by-side)

Form screens

- Create and edit use the **same page**
- Layout is always:

```
[ Save / Discard / Action buttons ]
[ Status bar (workflow steps) ]
[ Tabs (Details / Approval / etc) ]
[ Form fields (tab-scoped) ]      [ Chatter ]
```

- Autosave occurs **after record creation**, only when valid
- Status changes and actions post messages to chatter

Key Architectural Decision (Solved Problem)

❌ Do NOT assemble forms using page-level blocks like: - page.statusbar - page.tabs - page.container stacks

This caused: - Too many borders - Weird ordering - Tabs above headers - Non-Odoo behavior

✅ **Correct solution**:

Statusbar and tabs live inside the **form view header**, not the page.

This makes Odoo behavior automatic and foolproof for AI.

Current Manifest Capabilities (v1.3 + extensions)

Page header

```
"header": { "variant": "none" }
```

- Suppresses the big title card - Used on list + form pages

Form view header (Odoo-style)

```
"header": {
  "save_mode": "top",
  "auto_save": true,
  "auto_save_debounce_ms": 750,

  "primary_actions": [
    { "action_id": "action.req_start" },
    { "action_id": "action.req_submit" }
  ],

  "statusbar": { "field_id": "req.status" },

  "tabs": {
    "style": "lifted",
    "default_tab": "details",
    "tabs": [
      { "id": "details", "label": "Details", "sections": ["main"] },
      { "id": "approval", "label": "Approval", "sections": ["approval"] }
    ]
```

```
      }
   }
```

Renderer order (hard-coded, Odoo-correct): 1. Save / Discard + action buttons 2. Statusbar (workflow) 3. Tabs 4. Fields (filtered by active tab)

The status field is automatically hidden from the form fields.

Canonical Page Pattern (Used Everywhere)

List page

```
{
  "header": { "variant": "none" },
  "content": [
    { "kind": "view", "target": "view:entity.list" }
  ]
}
```

Form page

```
{
  "header": { "variant": "none" },
  "content": [
    {
      "kind": "record",
      "entity_id": "entity.xxx",
      "record_id_query": "record",
      "content": [
        {
          "kind": "grid",
          "columns": 12,
          "items": [
            { "span": 8, "content": [ { "kind": "container", "variant": "card",
"content": [ { "kind": "view", "target": "view:entity.form" } ] } ] },
            { "span": 4, "content": [ { "kind": "container", "variant": "card",
"content": [ { "kind": "chatter" } ] } ] }
          ]
        }
      ]
    }
  ]
}
```

Exactly **two cards**. No nesting madness.

Autosave Rules (Odoo-like)

- Autosave only triggers **after record has an ID**
- Autosave only fires when:
- form is dirty
- validation passes (`required_when`, etc)
- debounce timer completes

Validation Guards Added

- `form.header.statusbar.field_id` must match workflow `status_field`
- `form.header.tabs[].sections[]` must reference real section IDs
- Every section must appear in at least one tab (unless explicitly allowed)

These guards prevent AI from generating broken layouts.

What Has Been Implemented (Confirmed)

- Odoo-style form headers
- Native statusbar + tabs in form views
- Page header suppression
- Autosave
- Clean list/form renderers (no debug headings)
- Studio default templates updated to follow this pattern

Files touched recently: - `FormViewRenderer.jsx` - `AppShell.jsx` - `manifest_validate.py` - `MANIFEST_CONTRACT.md` - `main.py` (Studio templates)

Important Instruction for Future AI

⚠️ **DO NOT** build Odoo-style forms using page-level `statusbar` or `tabs` blocks.

✅ Always: - Put statusbar + tabs in the **form view header** - Keep pages thin - Keep forms authoritative

This is non-negotiable.

Current Status

System is now in a **good, stable architectural state**.

Next logical steps (optional): - Chatter event hooks (status changes, create events) - Workflow-aware action button ordering - Optional readonly mode for Approved records

---

End of context file.