# WYSIWYG Web Design Tool — MVP Implementation Plan & Exhaustive Requirements

This document consolidates the MVP implementation plan and the extended requirements into a single comprehensive specification. It includes architecture, state design, acceptance criteria, accessibility, error handling, export completeness, performance requirements, and governance/legal guardrails.

## 1) Executive Summary

**Goal:** Ship a visual, code‑free layout tool that outputs valid, production‑ready HTML/CSS using web‑native primitives (flow, Flexbox, optional CSS Grid, absolute/fixed when toggled).

**Guiding constraints:** - Rectangles/boxes only; no overlapping except when explicitly absolute/fixed. - What‑you‑see‑is‑what‑you‑get at each breakpoint. - WCAG 2.1 AA guidance integrated (basic checks) and zero “impossible” CSS states through the UI.

**Primary outputs:** HTML + CSS export, on‑canvas preview, code preview pane.

## 2) High‑Level Architecture

* **Frontend:** React 18 + Redux Toolkit + dnd‑kit.
* **Rendering:** DOM-based designer leveraging CSS for layout (fidelity guaranteed).
* **Styling:** CSS Modules (default) with build‑time Autoprefixer.
* **Code Editor:** Monaco Editor (read‑only with guarded refine mode).
* **Validation:** Lightweight rules engine for CSS/HTML checks; color contrast calc; keyboard traps; missing alt.
* **Undo/Redo:** Command system over Redux actions with history compaction.
* **No backend** for MVP beyond local storage import/export.

## 3) Core Domain Model (Redux state)

export type BreakpointId = 'mobile' | 'desktop' | 'large';  
  
export interface ProjectState {  
 meta: { id: string; name: string; createdAt: string; updatedAt: string; schemaVersion: number };  
 ui: UIState;  
 canvas: CanvasState;  
 rectangles: EntityState<RectangleNode>;  
 styles: StylesState;  
 components: ComponentsState;  
 breakpoints: BreakpointsState;  
 history: HistoryState;  
 export: ExportState;  
}

**Key elements:** - RectangleNode: boxes with layout, position, spacing, content, and per-breakpoint overrides. - StylesState: CSS classes with rules, specificity order, and palette tokens. - perBreakpoint: overrides applied by breakpoint.

## 4) Project File Format

* JSON serialization of ProjectState + CSS bundle string.
* Includes schemaVersion for migration safety.

## 5) Layout System

* **Split:** Shift + drag edge or context menu. Creates siblings, copies styles, enforces min sizes.
* **Merge:** Right‑click shared border or collapse drag. Combines nodes, merges styles.
* **Positioning:** Flow by default. Toggle absolute/fixed with handles and unit picker (%, px, rem, vw).
* **Visual feedback:** Hover outlines, guides, hierarchy breadcrumbs.

## 6) Content System

* **Text:** Inline editing, presets (Headline, Paragraph, Article, Lorem Ipsum), typography props.
* **Images:** Placeholders with aspect ratios, object-fit options, alt required.
* **Acceptance:** Content swaps never break layout; ratios preserved; text reflows across breakpoints.

## 7) Styling & Class Management

* **Live class generation** with semantic names.
* **Specificity control** via draggable list.
* **Granularity:** Basic (spacing, color, typography), Advanced (layout), Expert (guarded CSS editor).
* **Style palette:** character vs. component styles; apply as additive or replacement.
* **Conflict warning:** highlight property group conflicts.

## 8) Component System

* Built‑ins: Button, Link, Input (text/email/submit), Image, Text container.
* **States:** base, hover, focus, active, disabled via State Editor.
* **Preview mode:** toggle states for testing.

## 9) Responsive Design

* **Breakpoints:** Mobile (320–768), Desktop (769–1200), Large (1201+). Custom allowed.
* **Per‑breakpoint editing:** overrides stored separately.
* **Visibility toggle:** per‑breakpoint hide flag compiled to display:none.
* **Preview:** side-by-side comparison, live reflow toggle.

## 10) Export System

### 10.1 Structure

* /index.html
* /styles/main.css
* /assets/images/\*

### 10.2 HTML

* <div>, <p>, <img>; optional semantic tags post-MVP.
* ARIA attributes added only when required.

### 10.3 CSS

* Classes only, no IDs.
* One concern per class; deduped rules; media queries per breakpoint.
* Tokens via :root { --color-primary: ... }.

### 10.4 Formats

* HTML/CSS package, PNG/JPG snapshot, PDF via print CSS.

### 10.5 Validation

* Autoprefixer, minification, subset W3C checks.
* Block export if missing alt text, unlabeled input, or contrast failures.

## 11) Accessibility

* **Editor accessibility:**
  + Keyboard equivalents for all actions.
  + ARIA live for structural changes.
  + Respect prefers-reduced-motion and prefers-contrast.
* **Output accessibility:**
  + Color contrast check (4.5:1 / 3:1 large text).
  + Images require alt or decorative mark.
  + Inputs require labels.
  + Warn on skipped headings or missing landmarks.

## 12) Error Handling & Recovery

* Autosave to local storage every 10s and on blur.
* Crash recovery: load unsaved state with banner.
* Safe Mode disables Expert CSS on parse failure.
* Inline validation errors with actionable suggestions.

## 13) UI/UX Enhancements

* **Command Palette:** Ctrl/Cmd+K searchable action launcher.
* **Starter Templates:** Hero + 2 Columns, Blog List, Form Block.
* **Rulers & Units:** optional rulers, px + chosen unit display, clamp() helper for fluid typography.
* **Snap & Alignment:** snap to edges, distribute siblings evenly.

## 14) Performance

* CSS build debounced at 300ms, in Web Worker.
* DOM complexity warnings at depth >10 or children >12.
* Overlay virtualization for deep trees.
* History capped by memory (50 MB) with snapshot every 20 commands.

## 15) Undo/Redo System

* Command pattern wraps Redux actions.
* Continuous drags compressed into one history entry.
* Scoped history for component editing.
* Visualization as timeline with labeled actions.

## 16) Governance & Legal

* **Licenses:** List dependencies in About screen; export LICENSES.txt if required.
* **Privacy:** Entirely client‑side. No remote calls.
* **Telemetry:** Opt‑in only. Events: time‑to‑first‑layout, successful export, validation errors.

## 17) Testing Strategy

* **Unit tests:** reducers, selectors, CSS generation.
* **Component tests:** property panels, class order drag, state editor.
* **E2E:** Playwright flows with pixel diff per breakpoint.
* **Accessibility tests:** axe-core in preview; color contrast unit.
* **Testing matrix:** Chrome 90+, Firefox 88+, Safari 14+ (Mac+iOS), Pixel 5+ Android, Windows 10 laptop, MacBook Air M1, iPad Safari.

## 18) Non‑Functional Requirements

* Performance: <3s load, <100ms interaction.
* Reliability: autosave, crash recovery, schema migrations.
* i18n: MVP English only; externalized strings.
* Security: sanitize imports, block unsafe code.

## 19) Milestones (16 weeks)

* **Phase 1 (Weeks 1–4): Core Layout**
  + Rectangle graph + renderer; split/merge; selection; basic text/image; undo/redo; skeleton view.
* **Phase 2 (Weeks 5–8): Styling & Components**
  + Style palette; class order; built-ins; State Editor.
* **Phase 3 (Weeks 9–12): Responsive & Export**
  + Breakpoints; overrides; side-by-side view; code preview; export; validation.
* **Phase 4 (Weeks 13–16): Polish & Perf**
  + Performance monitors; UX refinements; crash recovery; onboarding.

## 20) Definition of Done

* All acceptance criteria for layout, styling, components, responsive, export, accessibility, undo/redo.
* Accessibility passes axe-core.
* Export parity validated within ±1px.
* Starter templates included.
* Recovery and telemetry tested.

**End of Document**