



Figma-Style Design Tool

DOM-Only Visual Editor (Foundation Version)

Project Goal

The objective of this project is to build a basic visual design editor similar in spirit to Figma, but implemented entirely using standard DOM elements. The emphasis is not on advanced rendering or performance optimizations, but on demonstrating a strong understanding of DOM manipulation, event handling, coordinate calculations, and state management using plain JavaScript.

This project must avoid canvas, SVG engines, or external frameworks. Everything should be achieved through HTML, CSS, and vanilla JavaScript.

Core Functional Requirements

1. Element Creation

The editor should allow users to create two types of elements:

- Rectangle
- Text box

Each created element should be represented by a simple `<div>` element in the DOM. When an element is added:

- It must be assigned a unique ID.
- It should appear with a default width, height, and position inside the canvas.
- The element type (rectangle or text) must be stored as metadata for later use.

2. Single Element Selection

Only one element can be selected at any given time.

Selection behavior should work as follows:

- Clicking on an element selects it.
- The selected element must visually indicate selection using a border or outline.
- Four resize handles should appear on the corners of the selected element.
- Clicking anywhere outside the element (on the canvas) should deselect the current element and remove the selection indicators.

Selection state must be centrally managed so that all tools (dragging, resizing, properties panel, keyboard shortcuts) operate on the same selected element.

Core Functional Requirements

3. Dragging, Resizing, and Rotation

Dragging:

- Elements should be draggable using the mouse.
- Movement should be calculated using mouse events and `offsetX / offsetY` or equivalent logic.
- Dragging must be constrained so elements cannot move outside the canvas boundaries.

Resizing:

- Resizing should be allowed only from the four corner handles.
- Width and height should update dynamically as the mouse moves.
- Minimum size constraints should be applied to avoid negative or zero dimensions.

Rotation:

- Elements should support basic rotation using CSS transforms.
- Rotation logic should be simple and event-based, not physics-driven or engine-based.
- Rotation state must be preserved when saving and loading.

4. Simple Layers Panel

A basic layers panel should be implemented as a vertical list showing all elements currently present on the canvas.

Features:

- Each list item represents one element.
- Clicking a layer selects the corresponding element on the canvas.
- Two buttons should be provided:
 - Move Up (bring element forward)
 - Move Down (send element backward)

Layer ordering must be reflected both visually (z-index) and in the internal data structure.

Core Functional Requirements

5. Basic Properties Panel

A properties panel should allow editing of limited attributes for the selected element.

Editable properties:

- Width
- Height
- Background color
- Text content (only for text elements)

Behavior:

- Inputs should be directly bound to the selected element.
- Any change in the input fields should immediately update the element's styles or content in real time.
- The panel should update automatically when a different element is selected.

6. Keyboard Interactions

Basic keyboard controls must be supported:

- Delete key removes the currently selected element from the canvas and internal state.
- Arrow keys move the selected element by 5 pixels per key press.
- Movement via keyboard must also respect canvas boundaries.

Keyboard handling should only apply when an element is selected.



Core Functional Requirements

7. Save and Load (Minimal Persistence)

The editor must support basic persistence using `localStorage`.

Requirements:

- The layout should be saved as a simple array of objects.
- Each object should contain all data required to reconstruct the element.

Example structure:

```
● ● ●  
{  
  id,  
  type,  
  x,  
  y,  
  width,  
  height,  
  styles  
}
```

On page refresh:

- The saved layout should be loaded automatically.
- All elements must be reconstructed with correct position, size, styles, and layer order.

Core Functional Requirements

8. Export Functionality

The editor should support exporting designs in two formats:

- JSON: Export the internal layout data as a downloadable JSON file.
- HTML: Export a basic HTML structure that visually reproduces the design using inline styles.

The export does not need to be production-ready, but it should accurately represent the current canvas state.

Instructions

Finish this project cleanly.

Write readable, well-structured code.

Add meaningful comments explaining intent and logic.

Avoid unnecessary complexity, but do not cut corners on correctness.