## 📘 ****Phase 1: Snap-to-Grid Isometric Foundation****

**Project Title:** HaikanCAD – Isometric Piping Grid with Precision Snap

### 📂 Table of Contents

1. **Overview**
2. **Objective**
3. **Core Components**
4. **Mathematics Behind Isometric Grid**
5. **Snap-to-Intersection Logic**
6. **UI & Interaction Behavior**
7. **Visual Proofs (Screenshots)**
8. **Key Source Files**
9. **Summary & Next Steps**

### 1. 📌 Overview

**HaikanCAD** is a custom-built CAD-like utility designed for industrial piping layout (配管図), developed with React and HTML5 Canvas. This Phase 1 milestone establishes a precise isometric grid and snapping cursor system, allowing users to interact with accurate grid intersections including slanted lines.

### 2. 🎯 Objective

* Create an **infinite isometric canvas** with light/dark mode
* Ensure grid consists of vertical and 30° slanted lines
* Accurately **snap the cursor** to **grid intersections**
* Lay the groundwork for future pipe/element placement

### 3. 🧱 Core Components

| File | Purpose |
| --- | --- |
| IsoGrid.jsx | Renders isometric grid (canvas) |
| SnapOverlay.jsx | Draws snapping crosshair (cursor) |
| App.jsx | Layout + grid toggle |
| App.css | Dark/light theming and layout |

### 4. 📐 Mathematics Behind the Grid

* **Spacing Unit:** spacing = 20px
* **Vertical lines:** Drawn every dx = 20
* **Slanted lines:** Drawn using Math.tan(π / 6) or 30° slope
* All calculations respect centered coordinate system:

js

CopyEdit

ctx.translate(canvas.width / 2, canvas.height / 2);

### 5. 🧠 Snap-to-Intersection Logic

#### Coordinate Conversion:

1. Translate mouse position to virtual grid space
2. Snap based on unit length and triangular geometry
3. Return to screen coordinates for visual display

#### Snapping Formula:

js

CopyEdit

// Convert screen to isometric

relX = mouseX - centerX

relY = mouseY - centerY

a = relX / dx

b = (relY \* 2 - relX) / dx

snapA = Math.round(a)

snapB = Math.round(b)

screenX = (snapA \* dx)

screenY = (snapA + snapB) \* dx / 2

### 6. 🖱️ UI & Interaction Behavior

* **Crosshair Cursor:** Red cross replaces system cursor
* **Mouse Movement:** Snaps to closest isometric grid corner
* **Responsive:** Canvas auto-resizes on window change
* **Debug Mode:** Optional background tint and logs

### 7. 📷 Visual Proofs

Include the following screenshots:

* ✅ Aligned crosshair at center dot
* ✅ Snapping to intersection (diamond tip match)
* ✅ Crosshair vs grid offset test (before fix)

### 8. 🧾 Key Source Files

#### IsoGrid.jsx

[✓ Draws full isometric grid]  
[✓ Red center dot for origin]

#### SnapOverlay.jsx

[✓ Snaps using true isometric projection]  
[✓ Crosshair always aligned with grid geometry]

(Refer to uploaded source snapshot for complete code.)

### 9. 🏁 Summary & Next Steps

You have now completed the **foundation for accurate isometric design tools**.

#### ✅ Achieved:

* Fully working infinite isometric grid
* Mouse snapping aligned to both vertical and slanted intersections
* Responsive and themed layout

#### 🛠️ Next:

* Tap-to-place or draw isometric segments
* User interaction tools (eraser, node label)
* Object anchoring (pipe segment start/end)
* Save/load JSON system
* Touch support for mobile/tablet