

Technical Test Assignment

Video Wall Size Calculator

[Loom here](#)

Objective

Leverage AI to build a working **Video Wall Size Calculator** that calculates the closest possible cabinet configuration (lower and upper) based on user-selected parameters.

The calculator must:

- Allow users to select cabinet type
- Allow users to input exactly **two parameters at a time**
- Calculate and display the closest **lower and upper** cabinet configurations
- Display number of rows and columns visually (simple grid diagram)
- Support unit conversion (mm, meters, feet, inches)

The goal is to replicate the logic described below.

Background Context

A **video wall** is built by stacking individual LED cabinets (like Lego pieces) horizontally and vertically until they form one large display.

Each cabinet has fixed physical dimensions.

The total video wall size depends on:

- Number of cabinets horizontally (columns)
 - Number of cabinets vertically (rows)
 - Cabinet type (16:9 or 1:1)
-

Cabinet Types

1) 16:9 Cabinet

- Width: **600 mm**
 - Height: **337.5 mm**
 - Aspect ratio: 16:9 (1.777...)
-

2) 1:1 Cabinet

- Width: **500 mm**
 - Height: **500 mm**
 - Aspect ratio: 1.0
-

User Inputs

The calculator must allow users to choose:

- Aspect Ratio (preset options provided — ignore custom aspect ratio)
- Height
- Width
- Diagonal

 The user may select **ONLY TWO parameters at a time.**

If two are selected:

- The other inputs must be disabled.
- If the user wants to change selection, they must unselect one first.

Example valid combinations:

- Aspect ratio + Height
- Aspect ratio + Width
- Height + Diagonal
- Width + Diagonal

Invalid:

- Selecting three or four parameters at once
-

Unit Toggle

The calculator must support:

- Millimeters
- Meters
- Feet
- Inches

When unit is changed:

- All entered values must convert automatically
 - All displayed results must update accordingly
-

Core Logic Requirements

The system must calculate:

- The closest possible cabinet configuration BELOW the requested size
- The closest possible cabinet configuration ABOVE the requested size

If an exact match is found:

- That match becomes the “size lower”
 - The next larger configuration becomes “size upper”
-

Example Scenario 1

Aspect Ratio: 16:9

Height: 100 inches

User clicks Apply.

The system must:

1. Convert 100 inches to mm
2. Determine how many cabinet rows are closest to that height
3. Calculate resulting total height
4. Check aspect ratio accuracy

5. Return:
 - Closest lower configuration
 - Closest upper configuration

Example output format:

Option 1 (Lower)

- 7 columns × 7 rows
- Total cabinets: 49
- Height: 93.01 inches
- Aspect ratio: 1.78

Option 2 (Upper)

- 8 columns × 7 rows
- Total cabinets: 56
- Height: 106.3 inches
- Aspect ratio: 1.78

The system must choose configurations that are closest in BOTH:

- Dimension accuracy
 - Aspect ratio accuracy
-

Example Scenario 2

Using 1x1 Cabinets

Target Aspect Ratio: 16:9

Height: 100 inches

Since 1x1 cabinets are square:

It may not be possible to achieve exact 1.78 ratio.

The system must:

- Calculate closest achievable ratio
 - Show that ratio (e.g., 1.8)
 - Still return closest lower and upper configurations
-

Example Scenario 3

Width + Diagonal

If user selects:

- Width: 100 inches
- Diagonal: 200 inches

The system must:

1. Compute implied height using geometry
 2. Determine closest cabinet grid combinations
 3. Return lower and upper matches for:
 - Width
 - Diagonal
-

Output Requirements

For each result option, display:

- Number of columns
- Number of rows
- Total cabinet count
- Final width
- Final height
- Final diagonal
- Final aspect ratio

Also display:

- The user's input values clearly
 - Whether the result is lower or upper
-

Visual Display

Below the results:

Display a simple grid diagram showing:

- Number of columns

- Number of rows

You do NOT need arrows or decorative elements.

A simple rectangular grid visualization is sufficient.

Behavior Rules

1. Only two inputs active at a time.
 2. Real-time unit conversion.
 3. Closest lower and upper results required.
 4. Exact matches handled properly.
 5. Must support both cabinet types.
 6. Must allow user to select configuration and confirm.
 7. Once selected, show final chosen dimensions clearly.
-

Edge Cases to Handle

- No exact aspect ratio possible
 - Very small sizes
 - Very large sizes
 - Decimal precision issues
 - Unit switching after results shown
-

What We Are Evaluating

We are testing:

- Logical thinking
 - Mathematical correctness
 - Use of AI to accelerate
 - Clean UI behavior
 - Edge case handling
 - Code structure
 - Accuracy of calculations
-

Deliverables

Please provide the following by end of day Tuesday:

1. A working hosted demo link
2. Source code repository
3. A Loom on how AI was used to accelerate
4. Short loom explanation of:
 - o Your calculation logic
 - o How you determine closest lower and upper
 - o How aspect ratio matching is handled
5. How long (in hours) it took you to build this

Email harish@goodspeed.studio, pinak@goodspeed.studio, edmond@goodspeed.studio

Important Notes

- Ignore custom aspect ratio option.
 - Ignore arrow graphics in UI.
 - Focus on accurate calculations and logic.
 - Clean UI is preferred but not primary focus.
-

If anything is unclear, document your assumptions and proceed logically.
