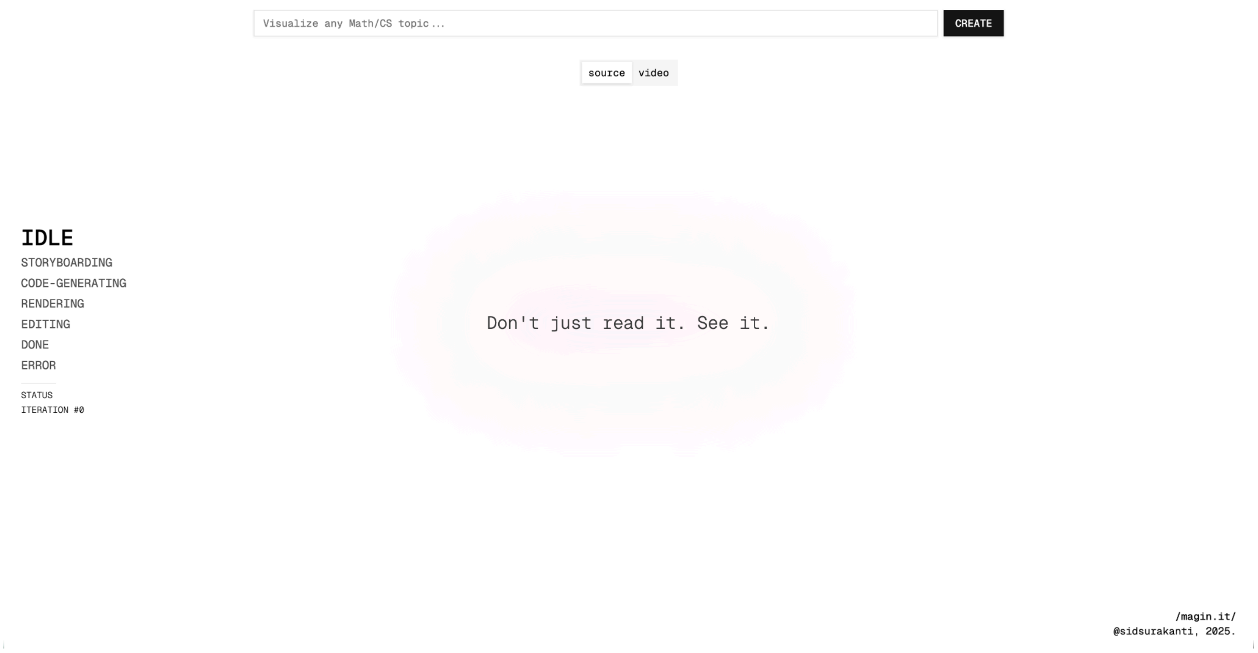
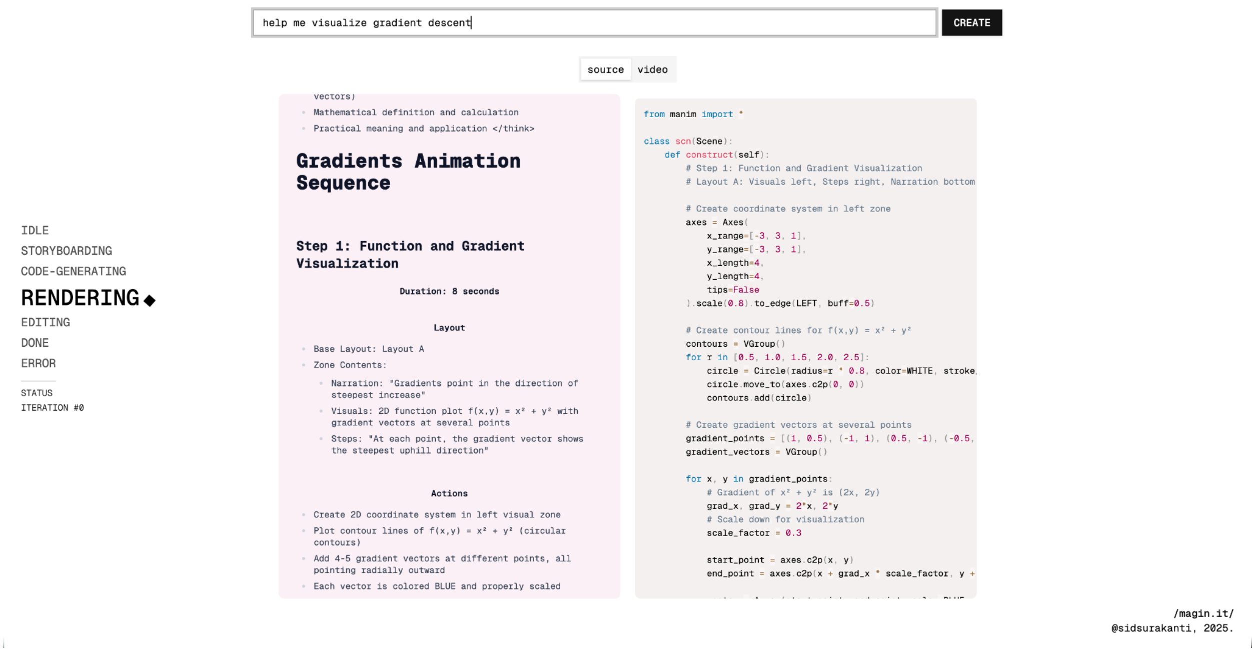




help me visualize gradient descent

vectors)

- Mathematical definition and calculation
- Practical meaning and application </think>

Gradients Animation Sequence

Step 1: Function and Gradient Visualization

Duration: 8 seconds

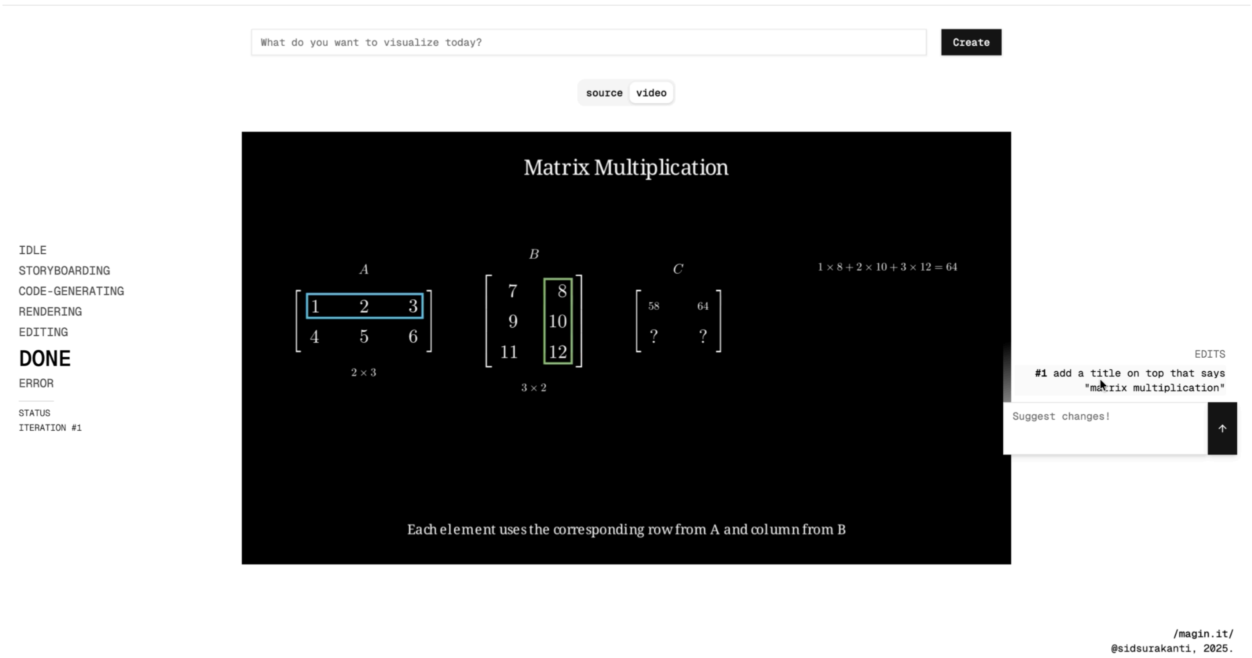
Layout

- Base Layout: Layout A
- Zone Contents:
  - Narration: "Gradients point in the direction of steepest increase"
  - Visuals: 2D function plot $f(x,y) = x^2 + y^2$ with gradient vectors at several points
  - Steps: "At each point, the gradient vector shows the steepest uphill direction"

Actions

- Create 2D coordinate system in left visual zone
- Plot contour lines of $f(x,y) = x^2 + y^2$ (circular contours)
- Add 4-5 gradient vectors at different points, all pointing radially outward
- Each vector is colored BLUE and properly scaled

```
from manim import *

class scn(Scene):
    def construct(self):
        # Step 1: Function and Gradient Visualization
        # Layout A: Visuals left, Steps right, Narration bottom

        # Create coordinate system in left zone
        axes = Axes(
            x_range=[-3, 3, 1],
            y_range=[-3, 3, 1],
            x_length=4,
            y_length=4,
            tips=False
        ).scale(0.8).to_edge(LEFT, buff=0.5)

        # Create contour lines for f(x,y) = x² + y²
        contours = VGroup()
        for r in [0.5, 1.0, 1.5, 2.0, 2.5]:
            circle = Circle(radius=r * 0.8, color=WHITE, stroke_
            circle.move_to(axes.c2p(0, 0))
            contours.add(circle)

        # Create gradient vectors at several points
        gradient_points = [(1, 0.5), (-1, 1), (0.5, -1), (-0.5,
        gradient_vectors = VGroup()

        for x, y in gradient_points:
            # Gradient of x² + y² is (2x, 2y)
            grad_x, grad_y = 2*x, 2*y
            # Scale down for visualization
            scale_factor = 0.3

            start_point = axes.c2p(x, y)
            end_point = axes.c2p(x + grad_x * scale_factor, y +
```



Matrix Multiplication

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad 2 \times 3$$

$$B = \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} \quad 3 \times 2$$

$$C = \begin{bmatrix} 58 & 64 \\ ? & ? \end{bmatrix}$$

$$1 \times 8 + 2 \times 10 + 3 \times 12 = 64$$

Each element uses the corresponding row from A and column from B

IDLE
STORYBOARDING
CODE-GENERATING
RENDERING
EDITING
DONE
ERROR

STATUS
ITERATION #1

EDITS

#1 add a title on top that says "matrix multiplication"

/magin.it/
@sidsurakanti, 2025.