

# Training Day 23 Report

Amrinder Singh

URN: 2302468    CRN: 2315013

15 July 2025

## Topic: CSS Grid – Advanced Layout Structuring

Today’s session focused on CSS Grid—a two-dimensional layout system designed to organize web content into precise rows and columns. Grid is perfect for building responsive designs, complex interfaces, and structured sections like galleries, dashboards, and landing pages.

## Key Areas Covered

### 1. Grid Container Setup

- Enabled grid behavior using `display: grid`.
- Defined columns and rows using `grid-template-columns` and `grid-template-rows`.
- Used values like `fr`, `auto`, and `px` for flexible sizing.

### 2. Grid Item Placement

- Placed items using `grid-column-start`, `grid-row-end`, and shorthand syntax.
- Applied `grid-area` for named layout regions.
- Explored implicit vs explicit grid tracks.

### 3. Alignment and Spacing

- Aligned items using `justify-items`, `align-items`, and `place-items`.
- Set grid gaps between rows and columns with `gap` and `grid-gap`.
- Combined grid with background and border styling for visual clarity.

### 4. Grid Use Cases Comparison

- Compared Grid to Flexbox: Grid excels in 2D layout, Flexbox in 1D flow.
- Common uses: image galleries, landing page sections, content blocks.
- Prepared for combining Grid and media queries in upcoming sessions.

## Hands-On Practice

Created a responsive pricing section with:

- Three pricing cards arranged in grid columns.
- Header and footer blocks aligned via grid rows.
- Adaptive behavior across mobile, tablet, and desktop views.

## Key Takeaways

- CSS Grid provides precise control over two-dimensional layout.
- Named areas and track sizing simplify complex designs.
- Combining Grid with Flexbox offers powerful responsive design options.
- Mastering Grid elevates layout capability to professional standards.