**Assignment 1:** Basic Grid Layout

**Task:** Create a 3x3 grid layout with equal-sized cells. Details:

Use **display: grid;**

Ensure all cells are of equal size.

**Visual Representation:**

A screenshot of a computer

Description automatically generated

**Assignment 2:** Nested Grids

**Task:** Create a 2x2 grid layout where each cell contains another 2x2 grid.

**Details:**

Use nested grid containers.

Ensure the inner grids are properly aligned within the outer grid.

**Visual Representation:**

A screenshot of a computer

Description automatically generated

**Assignment 3: Grid Areas**

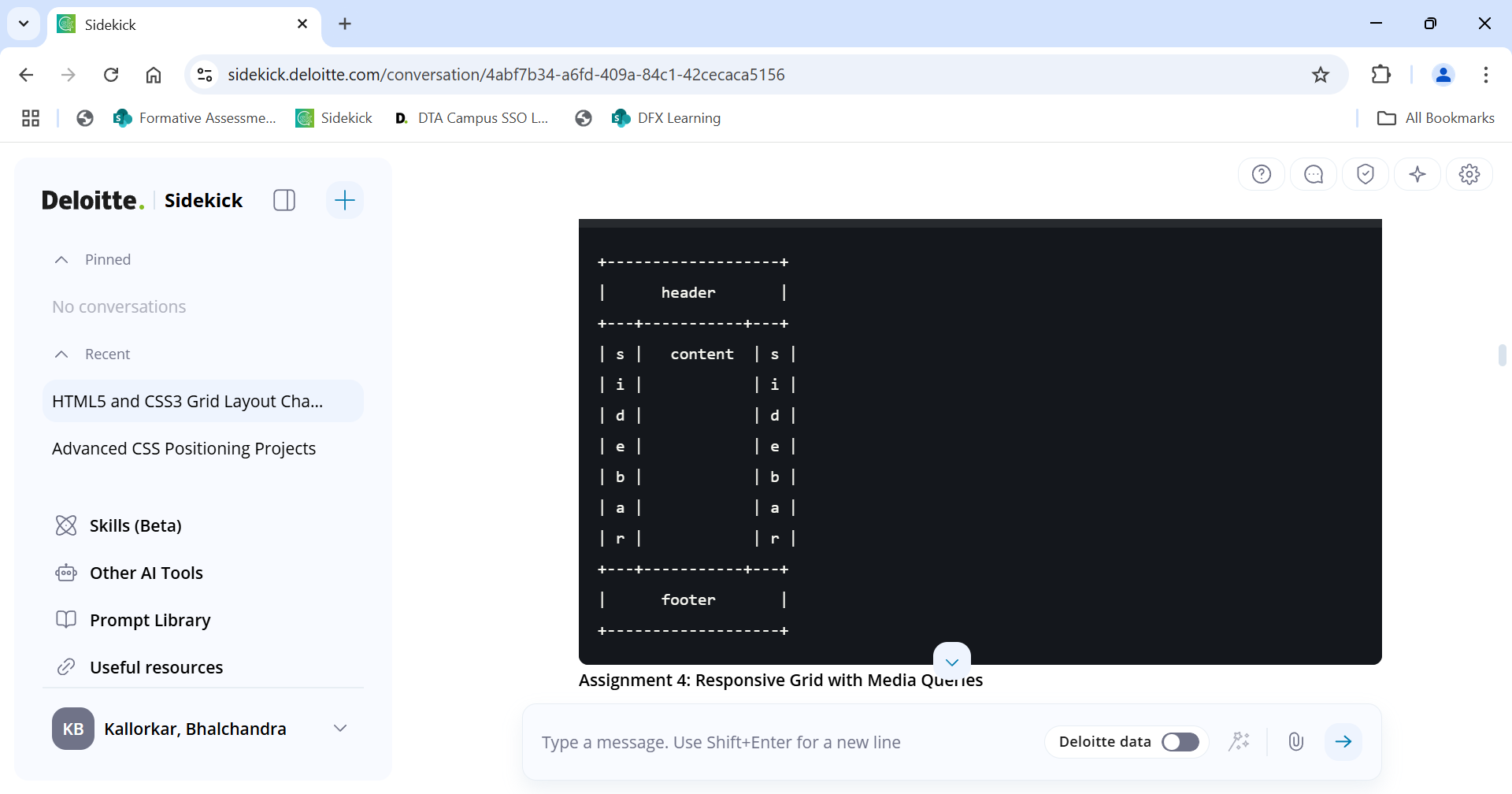
**Task:** Create a layout using named grid areas.

**Details:**

Define grid areas for header, sidebar, content, and footer.

Use grid-template-areas to arrange the layout.

**Visual Representation:**



**Assignment 4:** Responsive Grid with Media Queries

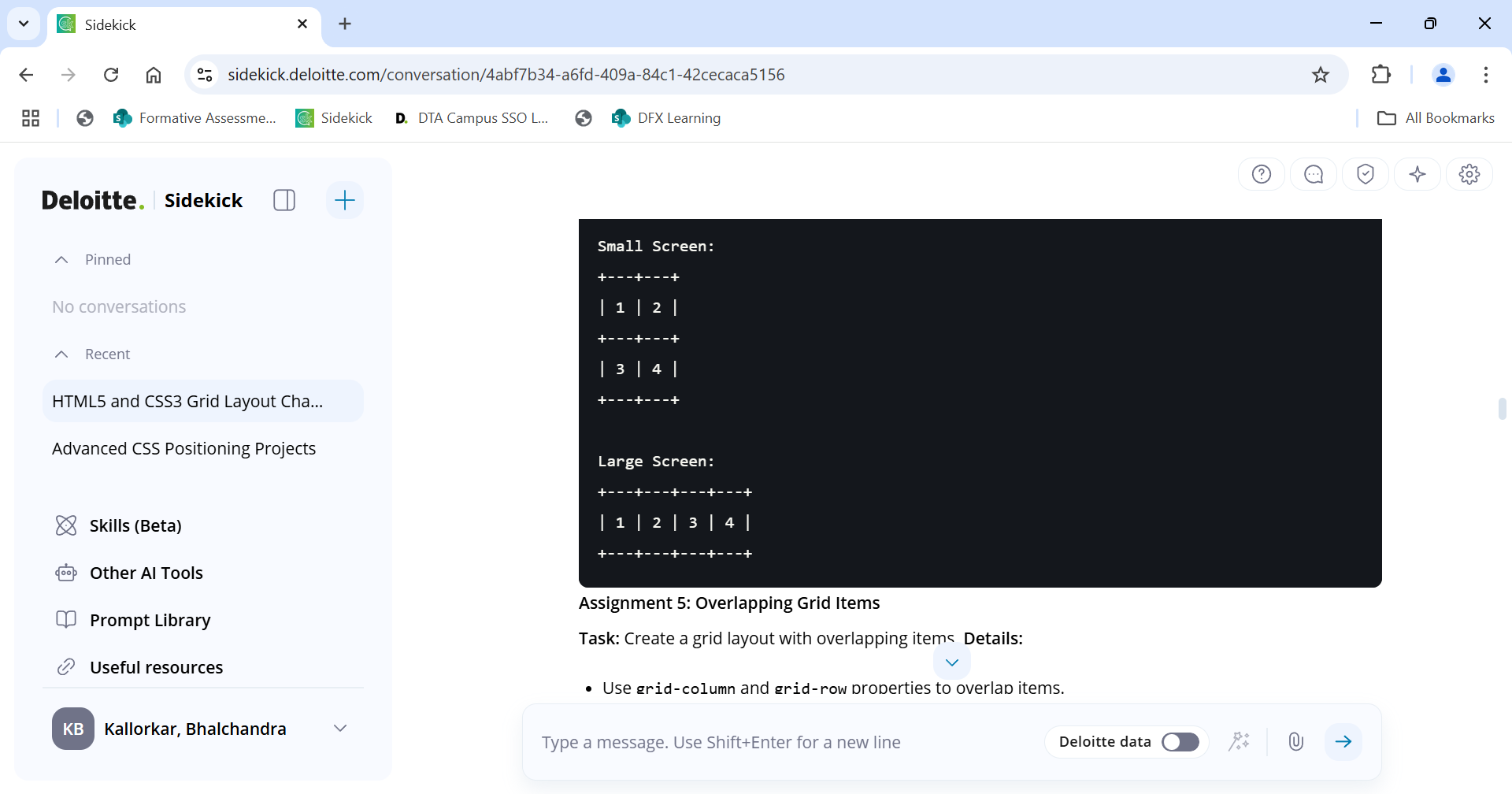
**Task:** Create a responsive grid layout that changes from 2 columns to 4 columns on larger screens.

**Details:**

Use media queries to adjust grid-template-columns.

Ensure the layout is fluid and adapts to screen size changes.

**Visual Representation:**



**Assignment 5:** Overlapping Grid Items

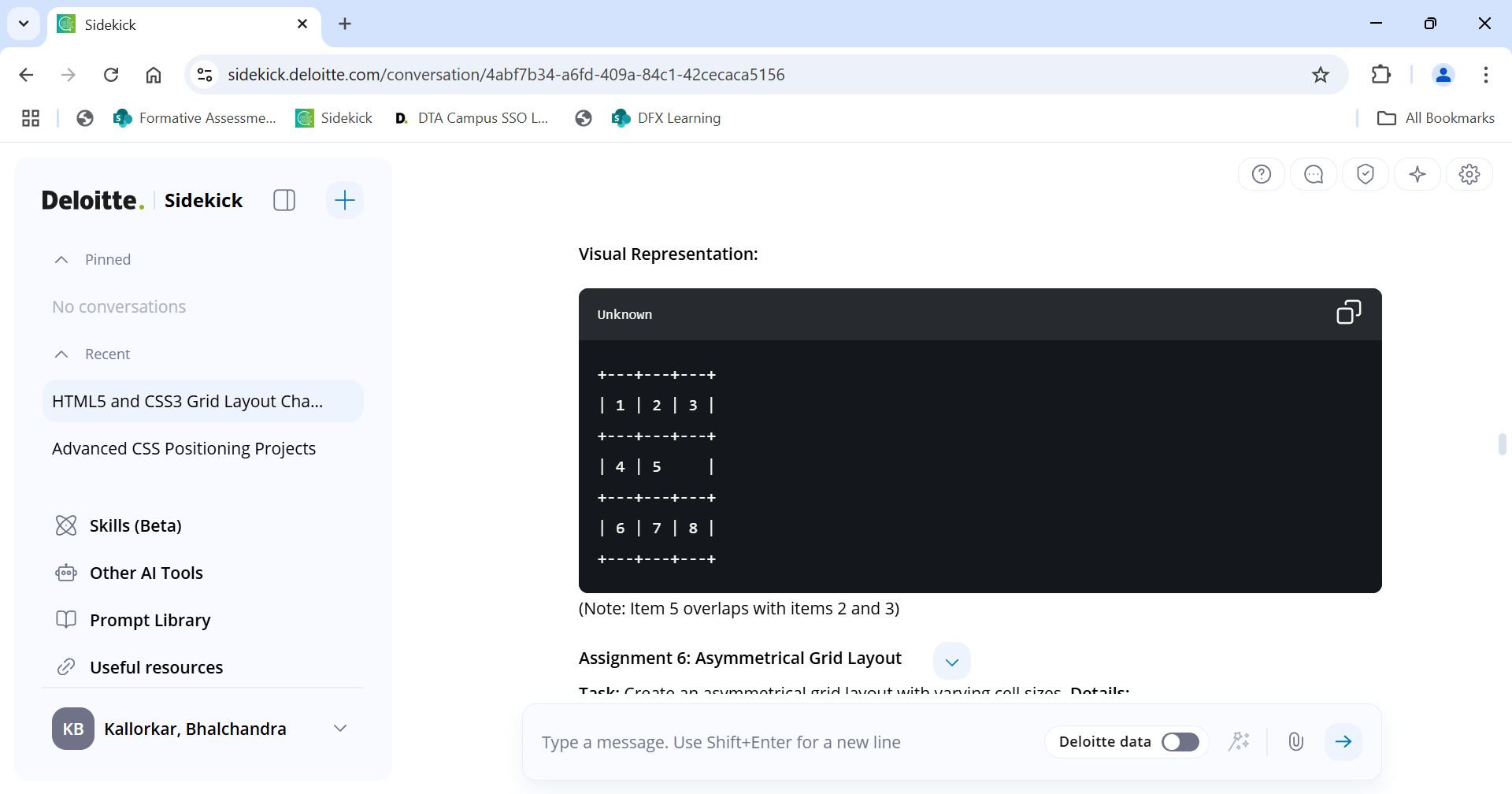
**Task:** Create a grid layout with overlapping items.

**Details:**

Use grid-column and grid-row properties to overlap items.

Ensure the layout looks visually appealing.

**Visual Representation:**



**Assignment 6:** Asymmetrical Grid Layout

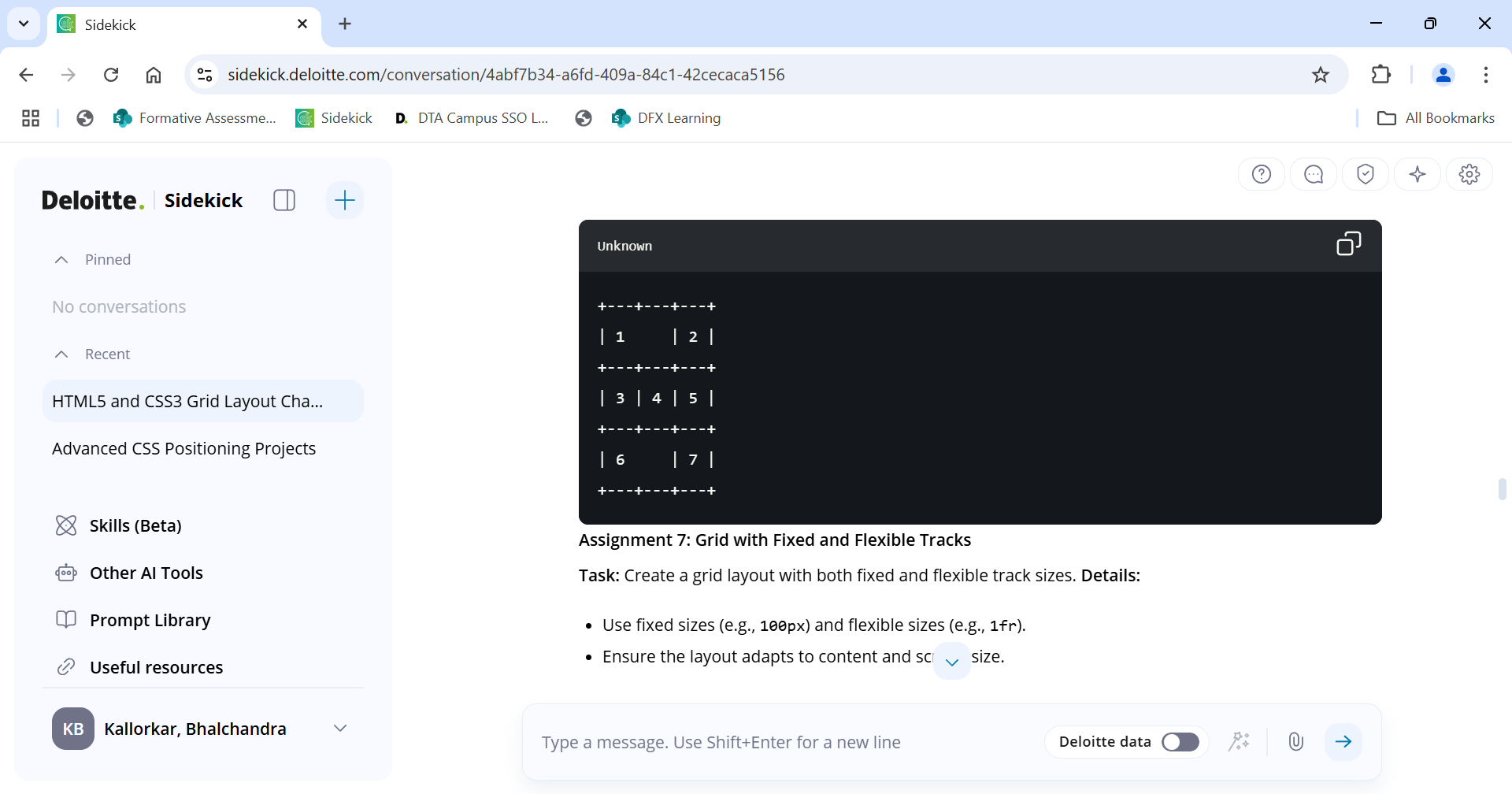
**Task:** Create an asymmetrical grid layout with varying cell sizes.

**Details:**

Use different grid-column and grid-row spans.

Ensure the layout is balanced and visually interesting.

**Visual Representation:**



**Assignment 7:** Grid with Fixed and Flexible Tracks

**Task:** Create a grid layout with both fixed and flexible track sizes.

**Details:**

Use fixed sizes (e.g., 100px) and flexible sizes (e.g., 1fr).

Ensure the layout adapts to content and screen size.

**Visual Representation:**

