Task 1 Questions...

Q1.What is the difference between a normal div and a grid container?
-> Normal div: items go one below another
Grid div: items can go in rows and columns

Q2.Why does the layout look the same even after applying Grid??
-> Because no rows or columns are set

Q3.Are grid items created automatically??
-> no, Only child elements become grid items

Q4.Can Grid work without rows or columns?
-> Yes, but no visible change

Q5.What do you think Grid needs to show visible changes?
-> Grid needs rows or columns to show change.

TASK 2: Grid Container – Columns

Task 2 Questions...

Q1. What does fr stand for?
→ **fr** stands for fraction of available space.

Q2. How many columns are created?
→ 3 columns are created because grid-template-columns: 1fr 1fr 1fr is used.

Q3. What happens when screen size is reduced?
→ The columns shrink automatically to fit the screen.

Q4. Do items overflow or adjust?
→ Items adjust, they do not overflow.

Q5. Why is fr better than px for layouts?
→ fr adjusts to different screen sizes, px does not.

Task1 | TASK 2: GRID C | TASK 3: FIX | TASK 4: GRID R | TASK 5: GRID G | TASK 6: GRID IT | TASK 7: GRID T | TASK 8: AUTO- | TASK 9: 12-CO | TASK 10: CSS G

File /Users/piyushowalekar/Library/CloudStorage/OneDrive-ITMTrust/HTML/Taks10/task3.html

Fixed Layout (px)

Fixed Box 1

Fixed Box 2

Fixed Box 3

Flexible Layout (fr)

Q1. Which layout breaks faster on small screens?
→ Fixed (px) layout breaks faster.

Q2. Which layout adjusts automatically?
→ Flexible (fr) layout adjusts automatically.

Q3. Why is fixed layout risky?
→ It does not adapt to screen size and may overflow.

Q4. Where can fixed columns be useful?
→ In tables, sidebars, and admin panels.

Q5. Which unit would you choose for a responsive website?
→ fr unit.

TASK 4: Grid Rows

Task 4 Questions...

Q1. What does auto do in grid rows?
→ auto adjusts height based on content.

Q2. Which row grows based on content?
→ The row with auto value grows.

Q3. Can rows have different heights?
→ Yes, each row can have different height.

Q4. What happens if content is very large?
→ auto row increases height to fit content.

Q5. Where are grid rows useful in real layouts?
→ In headers, content sections, and footers.

TASK 5: Grid Gaps

Q1. Where does gap add space?
→ Gap adds space between grid items.

Q2. How is gap different from margin?
→ Gap adds space between items, margin adds space outside items.

Q3. Does gap affect container size?
→ No, it only affects spacing between items.

Q4. Can we use gap only for rows or columns?
→ Yes, using row-gap or column-gap.

Q5. Why is gap preferred in Grid layouts?
→ It keeps spacing clean and controlled.

Task1 | TASK 2: GRID C | TASK 3: FIXED | TASK 4: GRID R | TASK 5: GRID G | TASK 6: GR × | TASK 7: GRID T | TASK 8: AUTO- | TASK 9: 12-CO | TASK 10: CSS G | +

File /Users/piyushowalekar/Library/CloudStorage/OneDrive-ITMTrust/HTML/Taks10/task6.html

TASK 6: Grid Item – Column Span

Q1. What are grid lines?
→ Grid lines are the vertical and horizontal lines that divide the grid.

Q2. How many columns does the item span?
→ The item spans 2 columns.

Q3. What happens to other items?
→ Other items move to the next available space.

Q4. Can two items overlap?
→ No, unless forced using positioning.

Q5. Where is column spanning used in websites?
→ In headers, banners, and featured sections.

TASK 7: Grid Template Areas

Questions & Answers

Q1. Why are template areas easy to understand?
→ Because they use names like header, sidebar, content instead of numbers.
Q2. Which section spans full width?
→ Header and Footer span full width.
Q3. Can we rearrange layout by changing area names?
→ Yes, we can change layout without changing HTML.
Q4. Is this better than column numbers?
→ Yes, it is easier to read and manage.
Q5. Where are template areas mostly used?
→ In website layouts, dashboards, and page structures.

TASK 8: Auto-fit, Minmax & Responsive Grid

Box 1

Box 2

Box 3

Box 4

Box 5

Box 6

Q1. What problem does auto-fit solve?
→ It automatically adjusts the number of columns based on screen size.

Q2. Why is minmax important?
→ It sets minimum and maximum width for grid items.

Q3. Do items shrink too much?
→ No, minmax prevents items from becoming too small.

Q4. Is this layout mobile friendly?
→ Yes, it adapts automatically to small screens.

Q5. Why are media queries not needed here?
→ Grid layout adjusts automatically using auto-fit and minmax.

TASK 9: 12-Column Grid System

Span 6 Columns	Span 3 Columns	Span 3 Columns

Q1. Why is 12-column grid popular?
→ Because it is easy to divide into halves, thirds, and quarters.

Q2. How does spanning help layout?
→ It allows elements to take more or less space as needed.

Q3. Where is this system used?
→ Used in websites, dashboards, and frameworks like Bootstrap.

Q4. Can we create layouts without this system?
→ Yes, but layout may be less consistent.

Q5. What happens if column count changes?
→ Layout structure changes and spans must be adjusted.

Task1 | TASK 2: GRID C | TASK 3: FIXED | TASK 4: GRID R | TASK 5: GRID G | TASK 6: GRID IT | TASK 7: GRID T | TASK 8: AUTO- | TASK 9: 12-CO | TASK 10: C

File /Users/piyushowalekar/Library/CloudStorage/OneDrive-ITMTrust/HTML/Taks10/task10.html

TASK 10: CSS @supports

TASK 10 Questions...

Q1. Why do we check browser support?
→ To make sure the browser supports CSS Grid before using it.

Q2. What happens if Grid is not supported?
→ The grid styles are ignored and normal layout is used.

Q3. Is this needed for all projects?
→ No, only needed when supporting old browsers.

Q4. Who uses this in real projects?
→ Developers who build websites for many browser versions.

Q5. What does safe CSS mean?
→ CSS that works without breaking layout in unsupported browsers.