@media (max-width: 960px) {

.app {

grid-template-columns: 1fr;

grid-template-rows: 64px auto 1fr;

grid-template-areas:

"topbar"

"sidebar"

"main";

}

}

1. @media (max-width: 960px)

This is a media query.

It says: “Only apply the following CSS when the screen is 960px wide or less.”

That usually means tablets and phones.

👉 Above 960px → use your normal layout (desktop).

👉 Below 960px → use this new layout (mobile).

2. .app { ... }

We’re overriding the grid layout of .app (the parent container) when on smaller screens.

3. grid-template-columns: 1fr;

Instead of multiple columns, we now have just one column that takes the full width.

Everything stacks vertically. 📱

4. grid-template-rows: 64px auto 1fr;

Defines 3 rows:

First row: 64px → reserved for topbar (fixed height).

Second row: auto → sidebar (height depends on its content).

Third row: 1fr → main area (takes up the rest of the space).

5. grid-template-areas

css

Copy code

"topbar"

"sidebar"

"main"

Each row now has a single area instead of multiple side-by-side areas.

The structure becomes:

diff

Copy code

+-------------------+

| topbar | 64px

+-------------------+

| sidebar | auto

+-------------------+

| main | fills rest

+-------------------+

🎯 Why This Matters

On desktop (wide screens): you had a sidebar next to the main content, with a topbar across the top.

On mobile (narrow screens ≤ 960px): everything stacks vertically — topbar, then sidebar, then main.

This prevents squished sidebars and makes the layout usable on phones.

👉 In short: this media query is your responsive breakpoint that transforms the dashboard-style layout into a mobile-friendly stacked layout.