STRETCHLEARN · WORKBOOK

Data Dashboard Design — Workbook

This workbook turns the course into a build. Alongside the four modules, you will write a dashboard brief, sketch a grid-based wireframe, audit and fix chart choices, design an accessible colour and KPI system, declutter a real table, and assemble a developer handoff. Use the templates to keep your decisions, specs, and tokens in one place so the finished dashboard is build-ready, not just pretty.

Foundations: What a Dashboard Is For

Define the decision, the audience, and the grid before you open any software.

Worksheet: One-Page Dashboard Brief

Choose a real or realistic dashboard you will design through this workbook. Fill in each field. The single decision sentence is the contract you will test every later choice against, so make it specific and outcome-focused.

Dashboard name

Single decision this dashboard supports (one sentence)

Primary audience (named role, not 'everyone')

Dashboard type (strategic / operational / analytical)

Refresh cadence (real-time / daily / weekly / monthly)

Question 1 the screen must answer

Question 2 the screen must answer

Question 3 the screen must answer

What is explicitly out of scope

Exercise: Squint-Test a Live Dashboard

Find an existing dashboard you can view (yours or a public example). Shrink it to a thumbnail or blur your eyes, then note what still stands out. Compare what dominates to what should dominate based on its likely primary decision.

• What single element stands out most at thumbnail size, and is that the intended headline?

• Where does your eye land first, and is that the top-left Z-pattern position?

• Which elements are emphasised (bold, coloured, large) that probably should not be?

• If you had two seconds only, could you answer the dashboard's main question?

Worksheet: 12-Column Wireframe Plan

Plan a grey-box wireframe on a 12-column grid. For each module on the screen, record where it sits, how many columns wide it is, and which question it answers. Keep the primary KPI module in the top-left. Use the wireframe-plan template to lay this out.

Module / tile name

Question it answers (from the brief)

Grid position (row, start column)

Width in columns (out of 12)

Hierarchy tier (primary / secondary / tertiary)

Chart or tile type planned

Checklist: Strategy and Grid Readiness

[ ] I have written a single decision sentence and a named primary audience
[ ] I have chosen one dashboard type and not mixed strategic, operational, and analytical needs
[ ] I have listed three to five questions the screen must answer
[ ] My primary KPI module sits in the top-left of the grid
[ ] Every tile edge aligns to a column line on the 12-column grid
[ ] Whitespace, not borders, is doing most of my grouping

Choosing the Right Chart

Match each question to the most accurate chart, and rebuild any misleading ones.

Worksheet: Question-to-Chart Mapping

For every question on your dashboard, name the relationship it expresses and the chart that answers it most accurately. Note the sort order and the baseline rule. Use the chart-selection template to record this and keep it as your build reference.

Question

Relationship type (comparison / trend / part-to-whole / distribution / correlation)

Chosen chart type

Sort order (descending value / chronological / none)

Axis baseline rule (bars start at zero? line baseline?)

Reference line or target to add

Exercise: Chart Autopsy and Rebuild

Take three charts from a real dashboard. For each, identify the visual encoding it uses (length, position, angle, area) and whether it misleads. Rebuild any chart that relies on angle, area, a dual axis, or a truncated bar baseline into a more accurate form.

• What encoding does each chart use, and where does it sit on the accuracy ranking?

• Does any bar chart use a non-zero baseline, and what does fixing it do to the story?

• Is there a pie with more than three slices that should become a sorted bar chart?

• Does any chart use two y-axes to imply a correlation that may not be real?

Worksheet: KPI Tile Design Sheet

Design each headline KPI tile so the number carries context. For every KPI, record the value formatting, the comparison, the direction-to-meaning mapping, and whether a sparkline is included. Use the KPI-tile template.

Metric name

BAN value format (e.g. 1.2M, 4.1%)

Comparison shown (vs target / vs prior period)

Is 'up' good or bad for this metric?

Colour and arrow tied to favourable change (yes/no)

Sparkline or mini-trend included (yes/no)

Time frame label

Checklist: Chart Integrity Check

[ ] Every bar chart starts its value axis at zero
[ ] Bars are sorted by value, not alphabetically, unless order is meaningful
[ ] Line charts are used only for time or another continuous axis
[ ] No pie chart has more than three clearly different slices
[ ] No chart uses dual independent y-axes to manufacture a correlation
[ ] Every KPI tile pairs its number with at least one piece of context

Colour, Tables, and Decluttering

Build an accessible colour system and strip the noise out of a real table.

Worksheet: Colour System and Accessibility Plan

Define your dashboard's colour roles and verify accessibility. Assign a hex value to each role, name the palette family where relevant, and record the second (non-colour) cue and the contrast check. Use the colour-system template.

Colour role (neutral base / accent / positive / negative / warning)

Hex value

Palette family if a scale (sequential / diverging / categorical)

Second non-colour cue (icon / sign / position / label)

Text contrast ratio vs background (target >= 4.5:1)

Passes colour-blind simulation (yes/no)

Exercise: Table Declutter Pass

Take one real table of about 10 columns. Strip every piece of non-data ink, fix alignment and number precision, then add exactly one quiet in-cell cue. Time how long it takes to answer a sample question before and after.

- Which gridlines, borders, and fills did you remove with no loss of information?
- How did you fix alignment and number precision per column?
- What single in-cell cue did you add (bar, heat shade, or one highlight)?
- Was the decluttered table faster to read on your sample question, and by how much?

Checklist: Colour and Clutter Audit

- [ ] I use one neutral base, one accent, and a small status set, not a rainbow
- [ ] The palette family matches the data (sequential for ordered, diverging for deviation)
- [ ] No status is conveyed by colour alone; every one has a second cue
- [ ] All text meets at least the WCAG AA 4.5:1 contrast ratio
- [ ] The design has been previewed in a colour-blind simulator
- [ ] Vertical gridlines and chartjunk have been removed from charts and tables

Building and Handing Off in Tableau, Power BI, and Figma

Translate the design into a build and a developer handoff that leaves no guesswork.

Worksheet: Build Setup Checklist Sheet

Record the build settings for your chosen tool so the dashboard inherits one consistent standard. Fill in the fixed size, grid settings, formatting baseline, and performance limits. Use the handoff-spec template's Build Setup sheet.

Tool (Tableau / Power BI)

Fixed canvas size (e.g. 1440 x 900)

Grid and snapping enabled (yes/no)

Workbook/theme-level number format

Gridlines and zero lines hidden (yes/no)

Max visuals per page (performance limit)

Exercise: Overview-First Interactivity Redesign

Take a dashboard that crams summary and detail onto one view. Redesign it so the default is the overview, with one filter, one drill-down path, and one rich tooltip carrying the detail that used to crowd the screen.

• What is the single overview the default state should answer for everyone?

• Which one filter and one drill-down path support the analysts who dig deeper?

• What detail moved from the resting view into a tooltip?

• How does a viewer always see the active filter and reset to default?

Worksheet: States and Edge-Case Spec

For each component, define how it renders in non-ideal conditions. These are the states most designs forget and the cause of most rework. Record the message or behaviour for each. Use the handoff-spec template's States sheet.

Component (KPI tile / chart card / filter bar / table)

Loading state behaviour

Empty / no-data state message

Error state behaviour

Long-label handling (truncate / wrap)

Negative, zero, and very large value rendering

Checklist: Handoff Completeness Check

[ ] Design tokens are defined for colour roles, spacing, and type, and referenced by name
[ ] Each component documents its variants and exact padding
[ ] Default, loading, empty, error, and no-data states are specified for every element
[ ] Filter behaviour, drill-down paths, sort orders, and refresh cadence are annotated
[ ] Edge cases (long labels, negatives, zero, huge numbers) have defined rendering
[ ] A developer who never spoke to me could build this dashboard from the spec alone

Your Action Plan

1. Write the one-page brief: decision sentence, named audience, dashboard type, refresh cadence, and three to five questions.
2. Sketch a grey-box wireframe on a 12-column grid with the primary KPI module top-left.
3. Map every question to its most accurate chart, set sort order, and force bar axes to zero.
4. Design KPI tiles with a BAN, a comparison, a favourable-change colour rule, and a sparkline.
5. Define a colour system (neutral base, accent, status set, each with a second non-colour cue) and verify contrast at least 4.5 to 1 and a colour-blind simulator pass.
6. Declutter every table and set one consistent type and number-formatting standard with tabular figures across the screen.
7. Build in Tableau or Power BI on a fixed canvas, grid on, single formatting baseline, and make it overview-first with one filter, one drill-down, and rich tooltips.
8. Define every component's loading, empty, error, and no-data states plus edge-case rendering, then assemble the Figma handoff with tokens, components, states, and annotations a developer can build from.