

**HAVISH M
CONSULTING**

HMC ONE
AI STRATEGY. **BUILT TO SCALE.**

Generative AI Playbook for Vibe Coding

Let's Talk
Let's Talk
Let's Talk
Let's Talk



NEHA JHA

HAVISH MADHUPATY



Generative AI Playbook for Vibe Coding

Havish Madhvapaty and Neha Jha

havishmconsulting.com/downloads

**DOWNLOAD ALL OUR
FILES + HANDBOOKS + CONTENT**



About Havish M Consulting

Havish M Consulting (HMC) is an award-winning analytics and AI advisory firm and a proud member of NASSCOM, recognized among SiliconIndia's Top 10 Data Analytics Companies of 2025. Since its founding, HMC has delivered over 2,500 corporate workshops — including sessions for Boards of Directors and C-suite leadership — across 45+ organizations spanning Energy, CPG, BFSI, Manufacturing, Retail, and Consulting, in 10+ countries across India, Middle East, UK and North America.

At the helm is Havish Madhvapathy, Founder of HMC and a leadership coach to Fortune 500 C-suites and Boards of Directors. He serves as visiting faculty at IIM Kozhikode, SRCC, and Masters' Union, and has taught internationally at NUS Business School and S P Jain School of Global Management. His work has been featured in Businessworld, The Economic Times, Entrepreneur, and Financial Express, and he has authored professional certification content with the Economic Times Masterclass series and for National Stock Exchange NCFM Pro Series. Havish was listed on BusinessWorld 40u40 Education, Analytics Insights 40u40 Analytics, Entrepreneur India and Indian Express Educator of the Year and has received Indian Achievers Award. He has also been nominated by Microsoft as an MVP.

Working alongside him is Neha Jha, Senior AI Consultant, who leads the day-to-day management of GenAI adoption journeys for HMC's enterprise clients. Neha is the connective tissue between HMC's frameworks and their real-world implementation — coordinating readiness surveys, cohort trainings, and rollout logistics across engagements with Amber Enterprises, Bharat Petroleum, RR Kabel, JK Cement and Landmark Group, among others. Her work ensures that HMC's adoption programs don't just launch well but sustain momentum long after the workshop ends.

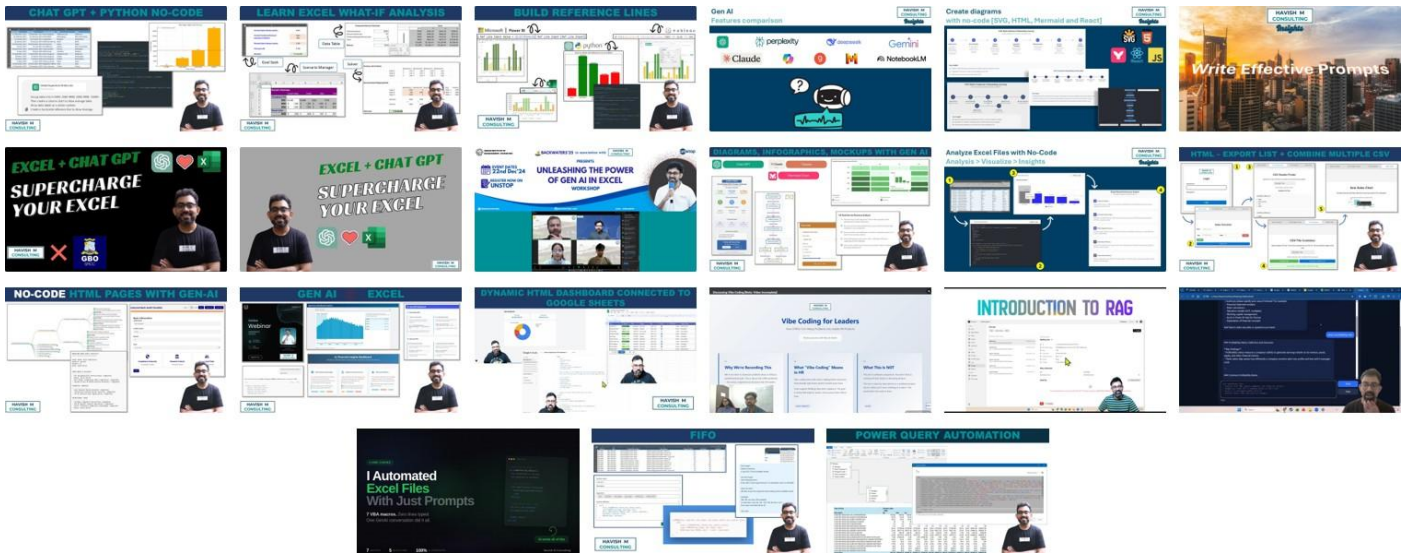
Backing this team is a strong advisory board, including Sanjay Kumar Suri (former CISO, Bharat Petroleum) and Jyotirmoy Bose (former CEO, Bennett Advisory) — bringing decades of enterprise security, governance, and strategic leadership experience to HMC's client engagements.

HMC's Generative AI adoption work sits at the heart of this book. The firm has created Generative AI labs for 8+ large organizations (USD 1 Billion +) guiding leadership teams from foundational awareness through functional training to enterprise-wide safety frameworks with measurable ROI.

YouTube

1600+ Videos covering Generative AI, Prompt Engineering, Agents, Excel Automation / Analytics / Dashboarding, Data Storytelling, No Code ML and more.

<https://www.youtube.com/c/havishmadhvapaty>



Key Clients



Client Testimonials



strategy&

Kushal Sinha

Partner @ Strategy&

“Havish adapts to audience requirements and enables the team to pick things up and run themselves. He covers advanced analytics techniques focusing on problem solving techniques.”



Yogesh Yadav

CFO at Shell Group Entity @ Sprng Energy

“Automation projects done helped us streamline and accelerate our workflows and enabled teams to complete complex tasks faster.”



IndusInd Bank

Siddharth Sinha

SVP and Head Analytics @ IndusInd Bank

“Wonderful session on GenAI. Really enjoyed”.



Rajeev Jain

SVP Marketing @ DS Group

“I have attended few training sessions from Havish on AI and enjoyed sessions thoroughly. I found him passionate about AI, having thorough knowledge and very sincere in approach. He understands need of trainees and customises sessions accordingly. I along with my team members learned a lot from him about applications of various AI tools.”



Rajwinder Singh

Head Equipment and Procurement LPG @ Bharat Petroleum

“Havish is an excellent trainer in Advanced Excel and Power BI. He is very competent and has the ability to respond quickly to any kind of queries. Had a great learning experience.”

GenAI Journey Snapshot across Clients

Client-anonymised view

Awareness = literacy and leadership buy-in at engagement start

Adoption = depth of embedded usage today

ROI = realised or tracked value

Industry	Sector	Awareness	Adoption	ROI
Wires & Cables	Electrical Equipment / Manufacturing	High at promoter and CHRO level; CIO governance concerns surfaced early	Growing — sanctioned enablement platform live; two flagship workstreams in sales planning and SCM	Medium-High — forecasting cycle compression, freight efficiency gains, reduced shadow-IT usage
Home Textiles, Pipes & Infrastructure	Textiles / Diversified Manufacturing	Strong promoter sponsorship across two distinct business lines	Function-deep rather than org-wide — planning and quality teams first	Medium-High — loom scheduling from days to hours; earlier defect capture; white-space led revenue pitches
Cement	Building Materials / Manufacturing	Concentrated in two business-unit heads; org-wide literacy still early	Executive-only by design — personal fluency before rollout	Medium — leadership self-sufficiency on analysis drafts; internal champions seeded ahead of scale
Retail (Fashion, Home, F&B)	Retail & Consumer	High at Gulf leadership level; category teams following	Mixed — leadership coaching alongside live pricing analytics	Medium-High — margin protection through better-timed markdowns; faster aged-stock clearance
Air Conditioners & Consumer Durables (OEM/ODM)	Consumer Durables / Manufacturing	Promoter-led from day one; baseline readiness survey segmented workforce into quadrant cohorts	Deep — enterprise licences at scale, always-on internal platform across plant, finance, commercial	High — analyst-hours reclaimed on BOM and variance reporting; board-level impact narrative funded Year 2
Steel & Power	Metals & Mining / Energy	Early-stage; discovery-led, building the	Pre-deployment — prioritised use-case pipeline, no	Early — avoided spend on low-value pilots; cross-functional

		business case before commitment	production rollout yet	consensus established
Pharmaceuticals	Healthcare / Life Sciences	Cautious but informed; governance framed the conversation before tooling	Selective — non- regulated functions only, use-case screening gate in place	Medium — faster market intelligence synthesis; reduced external research dependency
Banking	Financial Services	Function-led; strongest in Marketing, building in HR and IT	Broad but shallow — cross-functional training layer, workflow embedding next	Medium — lower agency dependency on first-draft creative; faster HR query turnaround
Trading, Retail & Real Estate	Diversified / Conglomerate	Leadership-driven; AI positioned as a strategy capability, not an IT one	Emerging — strategy and planning cell first, business units to follow	Medium — market scans in hours vs weeks; reduced first- pass consulting spend
Real Estate & Retail	Real Estate / Diversified	Engineering- function led; awareness strongest where estimation pain is felt	Targeted — design and estimation workflows in live use	Medium-High — faster bid-stage estimates; fewer revision cycles; senior engineer capacity freed
Power Transmission Infrastructure (EPC)	Power / Infrastructure	PMO-led; site teams onboarding progressively	Process-embedded — reporting and tracking layer across concurrent projects	Medium-High — multi-site status consolidation from days to hours; earlier delay flagging

Part — Basics

Article – Vibe Coding



Andrej Karpathy ✓
@karpathy



There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

3:17 PM · Feb 2, 2025 · 6.8M Views

'Vibe coding' named word of the year by Collins Dictionary

6 November 2025

Share

Save

Add as preferred on Google

Laura Cress

Technology reporter



Article – What is HTML

Why HTML Matters for Non-Coders

When you ask AI to build a dashboard, calculator or tool, it usually gives you an HTML file. If you use vibe coding, you will work with HTML often. You do not need to write it. You need to understand it well enough to check it, fix small things and ask for the right changes.

Why AI gives you HTML

HTML is the simplest way to share a working tool.

- It opens in any browser. No software to install.
- It works on laptops and phones.
- It can be emailed, saved to a shared drive or shared as a link.
- One file can hold text, tables, charts, buttons and calculations.

The three parts of an HTML file

Most HTML files have three parts:

- HTML: what is on the page. Headings, text, tables, buttons.
- CSS: how the page looks. Colours, fonts, spacing.
- JavaScript: what the page does. Calculations, filters, clicks.

Knowing this helps you give clear instructions. If the colours are wrong, that is CSS. If the totals do not update, that is JavaScript.



Why this matters if AI writes the code

Understanding HTML helps you in three ways.

- You can check the output. Text, labels and numbers are written in plain language inside the file. You can find and verify them.
- You can ask for better changes. Clear feedback means fewer rounds with AI.
- You can make small edits yourself. Changing a title, colour or number does not need AI.

When to use HTML

- Use Excel for data entry and analysis.
- Use PowerPoint for presentations.
- Use HTML when people need to use something more than once, click on it or open it on any device.

Simple rule: if people will read it once, use a document or slides. If they will use it again and again, use HTML.

Risks to know

- Anyone who opens the file can see everything inside it. Do not put salaries, customer data or pricing in a shared file.
- There is no login or access control.
- Many copies can spread by email. Keep one file in one shared location with one owner.
- A file that looks right can still be wrong. Test calculations against known numbers before using them.

How to get started

1. Ask AI to explain the file section by section in simple words.
2. Open the file in a text editor like Notepad. Find text and numbers you recognise.
3. Make one small change yourself. Save it and reopen it in the browser.
4. When asking for changes, say whether the problem is content, look or function.

Summary

AI can write HTML for you. Your job is to know what it is, check that it works, keep data safe and ask for clear changes. That is enough to turn AI output into tools your team can use.

Article – Understanding JSON

Why You Need to Understand JSON When Vibe Coding HTML

Vibe coding makes it possible to build surprisingly powerful HTML dashboards, tools, and mini-apps without writing every line of code yourself. You can describe what you want to ChatGPT, Claude, Copilot, or another AI coding assistant, and get a working interface within minutes.

But once your HTML page starts working with real data, one concept becomes extremely important: **JSON**.

JSON — JavaScript Object Notation — is essentially a structured way of storing and exchanging data.

Think of HTML as the **interface**, JavaScript as the **logic**, and JSON as the **data structure connecting everything together**.

For example, instead of hardcoding employee information inside an HTML page, you might store it like this:

```
[
  {
    "name": "Rahul",
    "department": "Sales",
    "sales": 1250000
  },
  {
    "name": "Priya",
    "department": "Finance",
    "sales": 850000
  }
]
```

Your HTML dashboard can then read this JSON and automatically create tables, charts, filters, KPI cards, or summaries.

Why JSON matters in vibe coding

The biggest reason is that **most modern applications exchange data in JSON format**.

When your vibe-coded HTML application connects to an API, AI model, database, workflow tool, or another application, the information will very often arrive as JSON.

For example:

Excel / CSV → JSON → HTML Dashboard

API → JSON → JavaScript → HTML

ChatGPT / Claude API → JSON → Your Application

n8n / Power Automate → JSON → HTML Interface

Even if AI writes the code for you, understanding the JSON structure allows you to understand **what the application is actually doing with your data**.

You do not need to become a JavaScript developer. But you should be comfortable recognizing concepts such as:

- Objects: { }
- Arrays: []
- Key-value pairs: "department": "Finance"
- Nested data
- Numbers, text, dates and Boolean values

This becomes especially useful when something goes wrong.

Imagine asking an AI assistant:

"Show sales by department."

The code might fail because your JSON uses "Dept" while the JavaScript is looking for "Department".

Without understanding JSON, this feels like a coding problem.

With basic JSON knowledge, you can immediately see that it is simply a **data structure mismatch**.

JSON also makes vibe-coded applications far more scalable.

Instead of writing:

Rahul - Sales - 12.5L

Priya - Finance - 8.5L

you store the data separately and let the application dynamically generate the interface.

Tomorrow, whether there are **10 records or 10,000 records**, the same HTML code can continue working.

This is the shift from building a static webpage to building a **data-driven application**.

The important mindset

Vibe coding does not mean that you need to understand every line of code generated by AI.

But you should understand the **building blocks**.

For most business users building HTML tools, dashboards, calculators, or internal applications, three concepts provide enormous leverage:

HTML tells the browser what to display.

JavaScript tells the application what to do.

JSON tells the application what data it is working with.

AI can write all three for you.

But once you understand JSON, you can give AI much better instructions, troubleshoot problems faster, connect more data sources, and build significantly more powerful applications.

That is why, for anyone learning vibe coding, **JSON is one of the highest-value technical concepts to understand.**

Article – Excel Upload Limit

If you have built a dashboard as a local HTML file, there is no standard Excel upload limit. A 200 MB workbook might load in one dashboard and fail in another. The answer depends on how the dashboard was coded, what the workbook contains, and how much memory the browser can use.

When you select a file in a local dashboard, the browser reads it from your computer. The page may have its own rule that rejects files above a certain size. If it does not, the next challenge is processing the workbook. Excel files can expand substantially in memory when the dashboard reads sheets, converts cells into data, and builds tables and charts. SheetJS notes that its standard reading functions load entire files into memory, where browser memory limits can become a constraint. ([SheetJS Large Datasets](#))

This is different from **Excel's worksheet limit** of 1,048,576 rows and 16,384 columns. That limit describes how much a worksheet can hold; it does not set the maximum file size a local HTML dashboard can accept. ([Microsoft Excel specifications](#))

If a 200 MB upload fails, first check whether the HTML code contains a file-size rule. If the page accepts the file but freezes or crashes, the likely issue is the work required to process and display it. Loading fewer sheets and columns, calculating summaries before drawing charts, and showing only a small preview of the rows can help. For consistently large files, preparing smaller or summarized data before opening the dashboard is often more reliable.

The practical answer: there is no universal “Excel upload limit” for a local HTML dashboard. Its working limit has to be tested with your actual file, browser, computer, and dashboard code. Converting the workbook to JSON alone does not guarantee that a large file will load.

Article – Working with Unclean Excel Files

Once you can generate a working HTML dashboard from a prompt, the bottleneck stops being the code. It becomes the file someone drags into it.

Almost every failure we see in these builds traces back to a spreadsheet that was designed to be read by a human and is now being read by a parser. Humans skip the title row, ignore the blank line, understand that the bold row at the bottom is a total. Code does none of this.

Twenty things to get right, on both sides — the file and the parsing logic.

Layout

- 1. Data starts at A1.** No company name in row 1, no logo, no "Report as of March 2025", no blank spacer row. The parser takes the first row it finds as the header. Give it the header.
 - 2. No merged cells anywhere.** A merged cell resolves to one value and a series of empty cells. Merged headers are the worst version of this, because the columns underneath end up unnamed and everything downstream is silently misaligned.
 - 3. One table per sheet.** Two tables side by side, or stacked with a gap, will be read as one rectangle with a lot of holes. If a sheet has a summary block above the detail, split them.
 - 4. Name the sheet you want.** Parsers default to the first sheet, which in most corporate workbooks is a cover page or an instructions tab. Read sheets by name in the code, and fail loudly with a list of available sheets when the name is missing.
 - 5. Hidden and filtered rows still come through.** Filters and hidden columns are a display state, not a data state. If someone filtered to Q4 before sending the file, the code still sees all four quarters. Delete, don't hide.
-

Headers

- 6. No two columns with the same heading.** Most parsers map rows to objects keyed by header. Two columns called "Amount" means one of them silently overwrites the other, and the dashboard will look completely plausible while being wrong.
- 7. No blank headings.** An unnamed column becomes `__EMPTY`, `__EMPTY_1`, and so on. Either name it or drop it before upload.
- 8. One header row, not two.** Stacked headers — a merged category row above a detail row — are a human convention. Flatten them into single names like `Q1_Revenue`, `Q1_Units` before the file goes anywhere near the parser.

9. Trim and normalise in the code. "Region " and "Region" are different keys, and the trailing space is invisible to whoever created the file. Strip whitespace, standardise case, and build a mapping layer rather than trusting the header string exactly as typed.

Data types

10. Numbers stored as text. The most common single cause of a chart that renders with everything at zero, or a total that concatenates instead of adding. Coerce explicitly and check what fails rather than assuming the type.

11. Currency symbols and separators live inside the cell. ₹1,20,000 is a string. So is \$4.5M. Format currency in the display layer, keep the underlying cell as a raw number.

12. Percentages are ambiguous. A cell showing 45% may hold 0.45 or 45 depending on how it was formatted. Decide the convention, state it, and validate against a known total.

13. Dates are the biggest trap. Excel stores them as serial numbers, so 45231 is a date that will parse as a number. Converting through a JavaScript date without controlling for timezone routinely shifts everything by one day. And 03/04/2025 is March or April depending on who made the file. Force an unambiguous format like YYYY-MM-DD at source.

14. Leading zeros disappear. Employee IDs, PIN codes, cost centre codes, account numbers. 007 becomes 7, and your join against another table quietly returns nothing. These are identifiers, not numbers — keep them as text.

Rows and content

15. No total or subtotal rows inside the data. A grand total row will be charted as a data point roughly twice the size of everything else. Totals are something the dashboard calculates, not something it inherits.

16. No blank rows in the middle, no footnotes at the bottom. * Figures exclude the Nagpur plant is not a data row, but it will become one. Strip trailing rows that don't match the column structure.

17. Blank is not zero. An empty cell and a 0 mean different things — one is "no sales", the other is "we don't know". Averages change materially depending on how you treat them. Decide deliberately, and say which you chose on the dashboard itself.

18. Category values must be consistent. North, north, North and NORTH will produce four bars in a chart that should have one. Normalise on read, and surface the distinct values so someone can spot the duplicates.

19. Never use a pivot table or a formula-heavy sheet as the source. Pivots are output, not data. Formula errors — #REF!, #N/A, #DIV/0! — flow straight through as strings and break every numeric operation after them. Paste as values into a clean sheet.

Practical

20. Reference columns by name, never by position. `row[3]` works until someone inserts a column, at which point the dashboard is wrong rather than broken — which is far more dangerous. Map by header name and throw an explicit error when an expected column is missing.

Two more worth building in as habit rather than rules. Browser-side parsing struggles well before you reach enterprise data volumes, so set a row limit and tell the user what it is instead of letting the tab freeze. And be deliberate about what goes into the file at all — a browser tool is exactly the wrong place to discover that a sheet has salary data or customer names in columns nobody was looking at.

The habit that prevents most of this

One sheet. One table. Starting at A1. One row of unique headers. No merges, no totals, no footnotes, no formatting doing the work of data.

Almost every rule above is a specific instance of that. Most teams get there faster by building a clean extract sheet next to the working file than by trying to reform the working file itself — the operational spreadsheet has a human audience and should keep it.

The parser gets its own copy.

Article – View HTML

<https://html.onlineviewer.net/>

The image displays a web browser window with a CSS preprocessor (Sass) code editor on the left and a preview of the generated CSS on the right.

Code Editor (Left):

```

1 <DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Xanban</title>
7 <style>
8 :root {
9   --page: #dde4ec; --soft: #eef2f7; --bg: #ffffff; --fg: #0f172a; --muted: #64748b
10   ; --line: #d8dce2; --faint: #f3f6f9;
11   --a: #2563b3; --onA: #ffffff; --good: #15803d; --warn: #b45309; --bad: #dc2626;
12   --c-todo: #64748b; --c-prog: #2563b3; --c-rev: #ea580c; --c-done: #16a34a;
13   color-scheme: light;
14 }
15 @media (prefers-color-scheme: dark) {
16   :root:not([data-theme="light"]) {
17     --page: #040810; --soft: #0a1120; --bg: #111a2e; --fg: #e8edf5; --muted:
18     #8fa0b8; --line: #22304a; --faint: #16203a;
19     --a: #60a5fa; --onA: #0b1220; --good: #4ade80; --warn: #fbbf24; --bad: #f87171
20     ;
21     --c-todo: #34a3b8; --c-prog: #60a5fa; --c-rev: #fb923c; --c-done: #4ade80;
22     color-scheme: dark;
23   }
24   :root[data-theme="dark"] {
25     --page: #040810; --soft: #0a1120; --bg: #111a2e; --fg: #e8edf5; --muted: #8fa0b8
26     ; --line: #22304a; --faint: #16203a;
27     --a: #60a5fa; --onA: #0b1220; --good: #4ade80; --warn: #fbbf24; --bad: #f87171;
28     --c-todo: #34a3b8; --c-prog: #60a5fa; --c-rev: #fb923c; --c-done: #4ade80;
29     color-scheme: dark;
30   }
31 }
32 *, ::before, ::after { box-sizing: border-box; }
33 html, body { height: 100%; margin: 0; overflow: hidden; background: var(--page); }
34 body { font-family: "Helvetica Neue", Helvetica, Arial, sans-serif; font-size:
35 14px; line-height: 1.4; color: var(--fg); }
36
37 /* Fixed 1600 x 900 (16:9) stage, scaled to fit any window */
38 .stage {
39   position: absolute; left: 0; top: 0;
40   width: 1600px; height: 900px;
41   transform-origin: top left;
42   background: var(--soft);
43   padding: 40px 40px 30px;
44   display: flex; flex-direction: column; gap: 12px;
45   overflow: hidden;
46   -webkit-print-color-adjust: exact; print-color-adjust: exact;
47 }
48
49 @page { size: 1600px 900px; margin: 0; }
50 @media print {
51   html, body { overflow: visible; }
52   .stage { transform: none !important; left: 0 !important; top: 0 !important; }
53 }
54
55 :focus-visible { outline: 2px solid var(--a); outline-offset: 2px; }
56 button, select, input { font: inherit; color: inherit; }
57
58 /* ----- Board ----- */
59 .board { flex: 1; min-height: 0; display: grid; grid-template-columns: repeat(4,
60 1fr); gap: 18px; }
61 .col { background: var(--bg); border: 1px solid var(--line); border-top: 4px solid
62 var(--cc); border-radius: 4px; display: flex; flex-direction: column; min
63 -height: 0; transition: background .12s; }
64 .col.over { background: var(--faint); outline: 2px dashed var(--cc); outline
65 -offset: -6px; }
66 .colh { display: flex; justify-content: space-between; align-items: center;
67 padding: 14px 16px 10px; }

```

Preview (Right):

The preview shows a grid-based layout with various components. The top navigation bar is dark with a light background. The main content area is light with a dark background. The sidebar is dark with a light background. The components include a sidebar, main content area, and a top navigation bar.

Article – Publish on GitHub

Publishing your HTML on GitHub

You have built something. A dashboard, a calculator, a one-page explainer. It sits in your Downloads folder as a single .html file, and it works perfectly when you double-click it.

Then someone asks you to send it, and the problem starts. Email strips it or flags it. WhatsApp won't take it. You put it on Drive and the recipient gets a download prompt and a file their browser refuses to open properly. The thing you made is trapped on your own machine.

What you need is a link. GitHub Pages gives you one, free, in about five minutes.

What GitHub Pages actually is

GitHub is where the world's code lives. Pages is a feature buried in its settings that says: take the files in this folder and serve them as a website. No server to rent, no hosting bill, no deployment pipeline. You put an HTML file in a folder, flip a switch, and it has a public address.

This works because the files you have been vibe coding are *static* — everything happens inside the visitor's browser. There is no database, nothing running on a server. That is precisely the kind of thing Pages was built to host.

The five-minute version

1. Make an account at github.com. Free is enough.

2. Create a repository. Top-right + → **New repository**. A repository — "repo" — is just a folder. Name it something short and lowercase, like sales-dashboard. Set it to **Public**. Tick **Add a README file**. Create.

On a free account, Pages only works on public repositories. Anyone who finds the URL can see your page, and anyone on GitHub can read the file. Paid tiers (Pro, Team) allow private repos with Pages.

3. Upload your file. On the repo page, **Add file** → **Upload files**. Drag your HTML file in. Scroll down, click **Commit changes**. No command line involved.

4. Turn Pages on. Go to the **Settings** tab → **Pages** in the left sidebar (under "Code and automation"). Under **Build and deployment**, set Source to **Deploy from a branch**, pick branch **main** and folder **/ (root)**, and Save.

5. Wait a minute, then refresh. A green banner appears with your URL. It will look like:

<https://yourname.github.io/sales-dashboard/>

GitHub says publishing can take up to ten minutes. It is usually under one. Send the link.

The one rule that trips everyone up

Name your file index.html.

index.html is what a web server looks for when someone visits a folder. If your file is called dashboard_final_v3.html, the bare URL shows an error or an ugly file listing, and the actual page only loads at .../dashboard_final_v3.html. Rename it before you upload and this problem disappears.

Two related gotchas: web servers are **case-sensitive** in a way Windows is not, so Index.html is a different file from index.html. And **avoid spaces** in file names — they turn into %20 in the URL.

Updating it

Upload the new version of index.html over the old one. Same URL, new content, live in about a minute.

If you don't see your change, your browser has cached the old page. Hard refresh with **Ctrl+Shift+R** (Cmd+Shift+R on Mac) before you assume something broke.

Before you click publish

A public repo is genuinely public, and "anyone with the link" includes search engines.

Open your HTML file and look at the data you hardcoded into it. If your demo dashboard contains a client's actual sales numbers, real customer names, or internal margins, that is now on the open internet with your name attached. Replace it with dummy data first. Same for API keys — a key pasted into client-side HTML is visible to anyone who views the source, and will be scraped within hours.

What this gives you, and what it doesn't

You get a real, permanent, shareable URL that works on any device and opens instantly — good enough for a client demo, a handbook example, a tool your team uses daily. You can point your own domain at it later if you want.

You don't get anything that needs a server: no login, no saving data between visits, no hiding anything behind a password. Those need a different class of tool.

For 90% of what you will build in this handbook, Pages is the whole answer.

havishmad / 3dmodel

CodeIssuesPull requestsAgentsActionsProjectsWikiSecurity and qualityInsightsSettings

3dmodelPublic

PinWatch0Fork0Star0

main1 Branch0 Tags

Go to fileTAdd fileCode

havishmadAdd files via upload✓

19519de · 2 weeks ago6 Commits

README.mdInitial commit2 weeks ago

index.htmlAdd files via upload2 weeks ago

README

3dmodel

About

No description, website, or topics provided.

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Deployments5

github-pages2 weeks ago

havishmad / 3dmodel

CodeIssuesPull requestsAgentsActionsProjectsWikiSecurity and qualityInsightsSettings

General

Access

Collaborators

Moderation

Code, planning, and automation

Rulesets

Branches

Tags

Actions

Webhooks

Copilot

Planning

Environments

Codespaces

Pages

Security and quality

Advanced Security

Deploy keys

Secrets and variables

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://havishmad.github.io/3dmodel/>

Last deployed by havishmad 2 weeks ago

Visit siteUnpublish site

Build and deployment

Source

Deploy from a branch

Branch

Your GitHub Pages site is currently being built from the main branch. [Learn more about configuring the publishing source for your site.](#)

main / (root) Save

Learn how to [add a Jekyll theme](#) to your site.

Your site was last deployed to the [github-pages](#) environment by the [pages build and deployment](#) workflow.

[Learn more about deploying to GitHub Pages using custom workflows](#)

Custom domain

Custom domains allow you to serve your site from a domain other than [havishmad.github.io](#). [Learn more about configuring custom domains.](#)

SaveRemove

Enforce HTTPS

— Required for your site because you are using the default domain ([havishmad.github.io](#))

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site.

When HTTPS is enforced, your site will only be served over HTTPS. [Learn more about securing your GitHub Pages site with HTTPS.](#)

Visibility

GitHub Enterprise

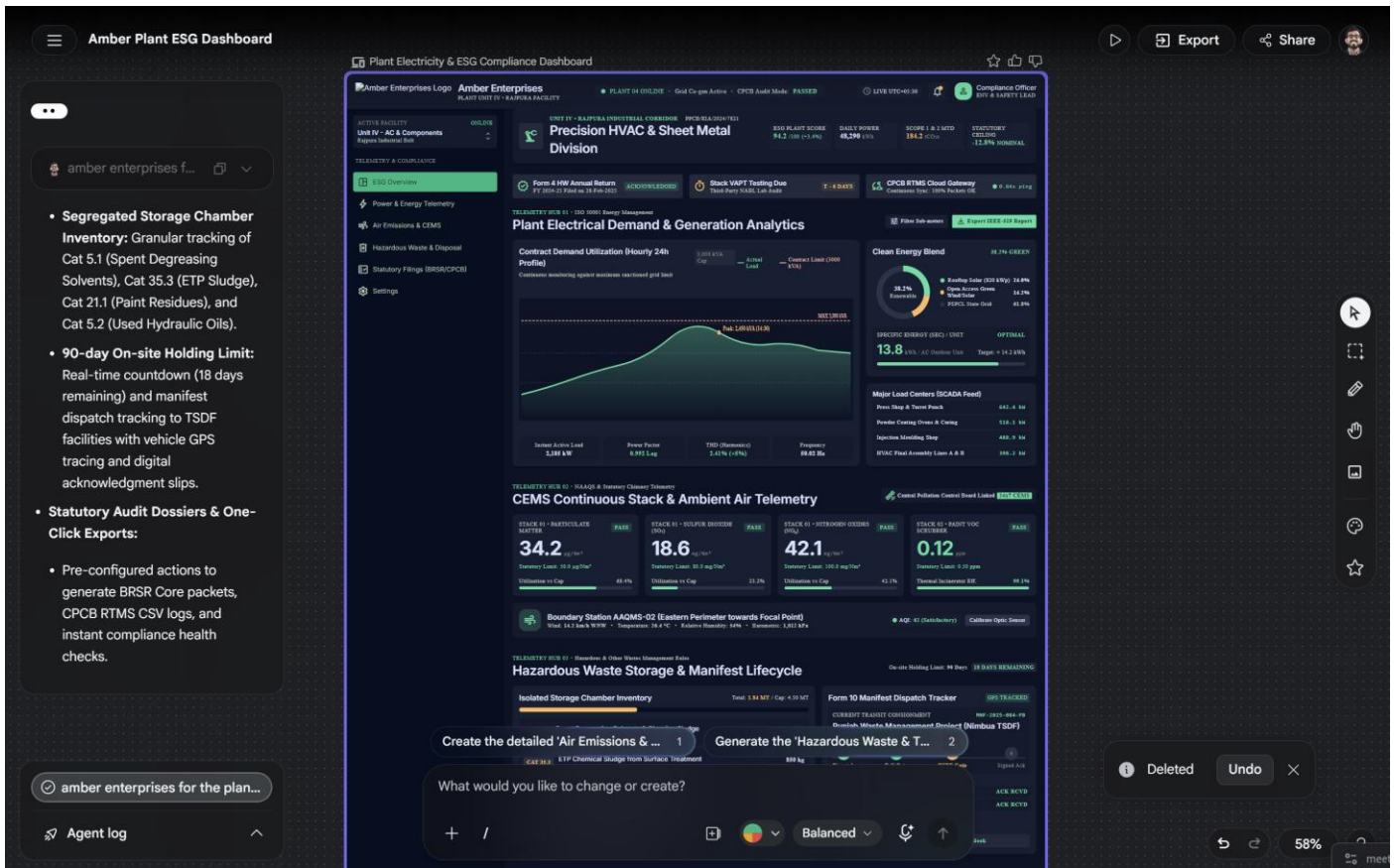
With a GitHub Enterprise account, you can restrict access to your GitHub Pages site by publishing it privately. You can use privately published sites to

Part — Dashboard and Reports

Article – Create Prototypes

Google Stitch

<https://stitch.withgoogle.com/>



Article – Slicers

1. Tile (button) slicer

Multi select

North

South

East

West

Selected: All

2. Dropdown slicer

Single select

All

Selected: All

3. List slicer

Single select, click again to clear

Electronics

Furniture

Apparel

Grocery

Selected: Electronics

4. Checkbox slicer

Multi select with Select all

☐ Select all

☐ Retail

☐ Wholesale

☐ Online

☐ Export

Selected: All

5. Chip slicer

Multi select

New

Bestseller

Sale

Premium

Eco

Limited

Selected: All

6. Toggle (switch) slicer

On / off filters

In stock only

Discounted

Include returns

On: None

7. Segmented slicer

Single select

Monthly

Quarterly

Yearly

Selected: Monthly

8. Range (between) slicer

Minimum and maximum

0

100

Price between: 20 and 70

9. Date range slicer

From and to

From

01-01-2026

To

21-09-2026

Period: 1 Jan 2026 to 21 Sept 2026

10. Search slicer

Type to filter

Type a customer name

Aarav Shah

Ananya Iyer

Bhavin Patel

Charu Mehta

Selected: All

Article – D3js Charts

<https://d3js.org/>



Q Search Ctrl K

7.9.0 GitHub 113.0k ★

Made by Observable

The JavaScript library for bespoke data visualization

Create custom dynamic visualizations
with unparalleled flexibility



Get started

What is D3?

Examples



▶ Play



2020

Speed

0.5x

1x

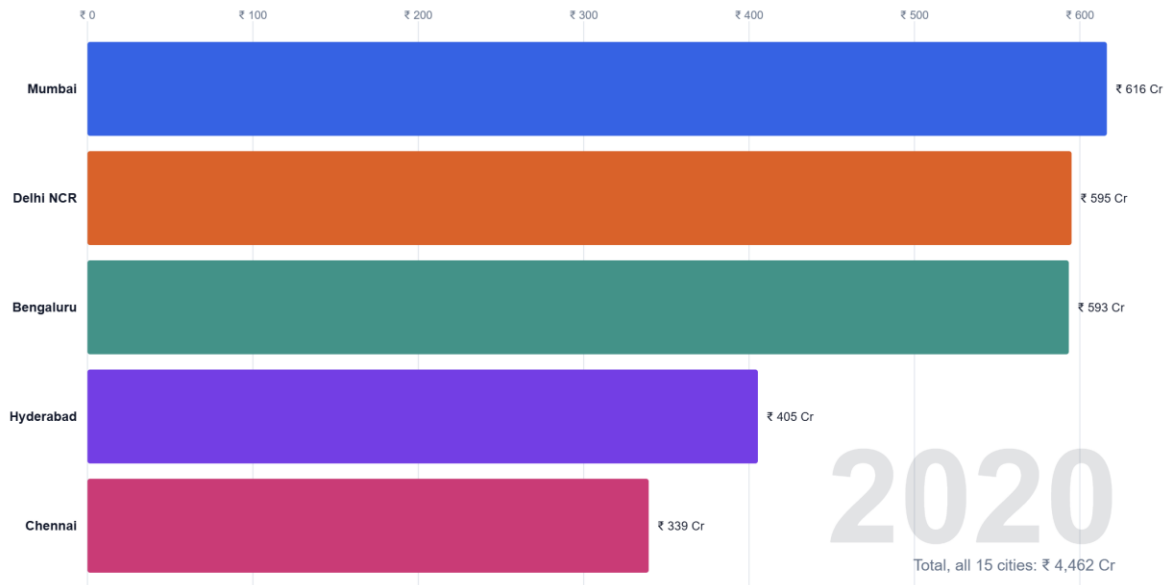
2x

Show

Top 5

Top 10

All 15



2020

Total, all 15 cities: ₹ 4,462 Cr

Annual revenue by city, ₹ Cr, 2015 to 2025. Press Play, or drag the slider to any point in time. Hover a bar for details.

Sample data - Built with D3.js

Article – Plotly Charts

<https://plotly.com/>

Claude + Plotly Cloud, plus major updates to plotly.py and plotly.js. [See what's new, live.](#)

Product ▾ExamplesCommunity ▾Enterprise ▾PricingResources ▾Company ▾

Sign In

Get Demo

Plotly builds the enterprise platform and open-source frameworks to deliver internal apps and analytics with AI coding agents

Request a demo →

Try our embeddable AI →

Trusted by the world's largest organizations





To exit full screen, press and hold Esc

RevenueGross profitOperating profitOperating margin

₹ 1,248 Cr₹ 506 Cr₹ 168 Cr13.5%

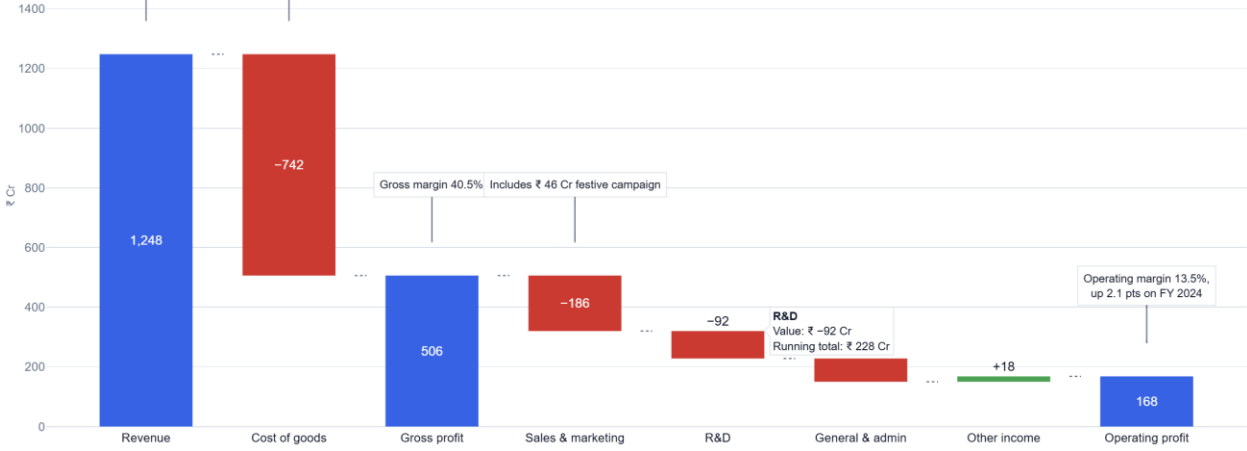
Replay

Speed0.5x1x2x

NotesOnOff

Up 12.4% on FY 2024

59.5% of revenue, 1.2 pts better than FY 2024



Category	Value (₹ Cr)	Running Total (₹ Cr)
Revenue	1,248	1,248
Cost of goods	-742	506
Gross profit	506	506
Sales & marketing	-186	320
R&D	-92	228
General & admin	-92	136
Other income	+18	154
Operating profit	168	168

Gross margin 40.5%

Includes ₹ 46 Cr festive campaign

Operating margin 13.5%, up 2.1 pts on FY 2024

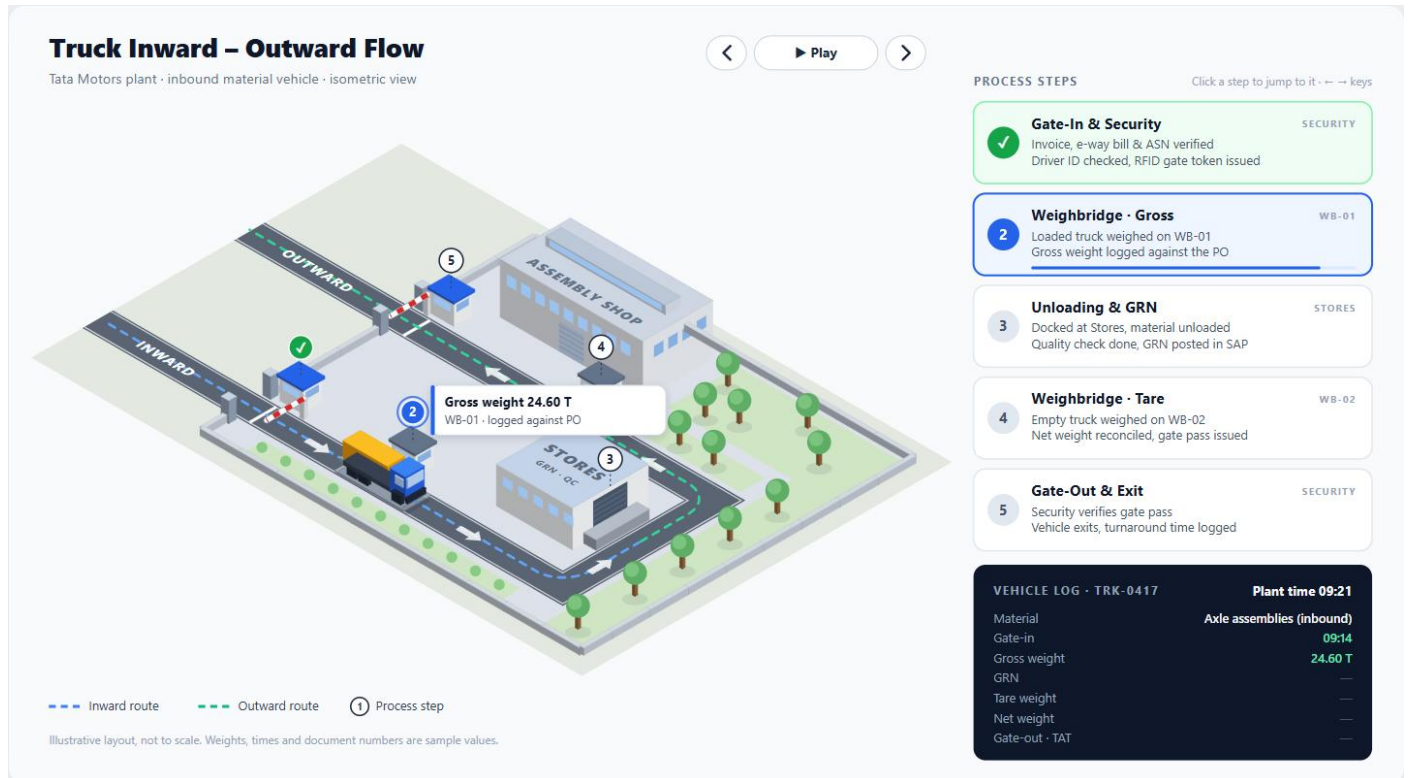
Revenue to operating profit, ₹ Cr, FY 2025. Bars build in sequence; hover any bar for its value and the running total.

Sample data · Built with Plotly.js

Article – SVG

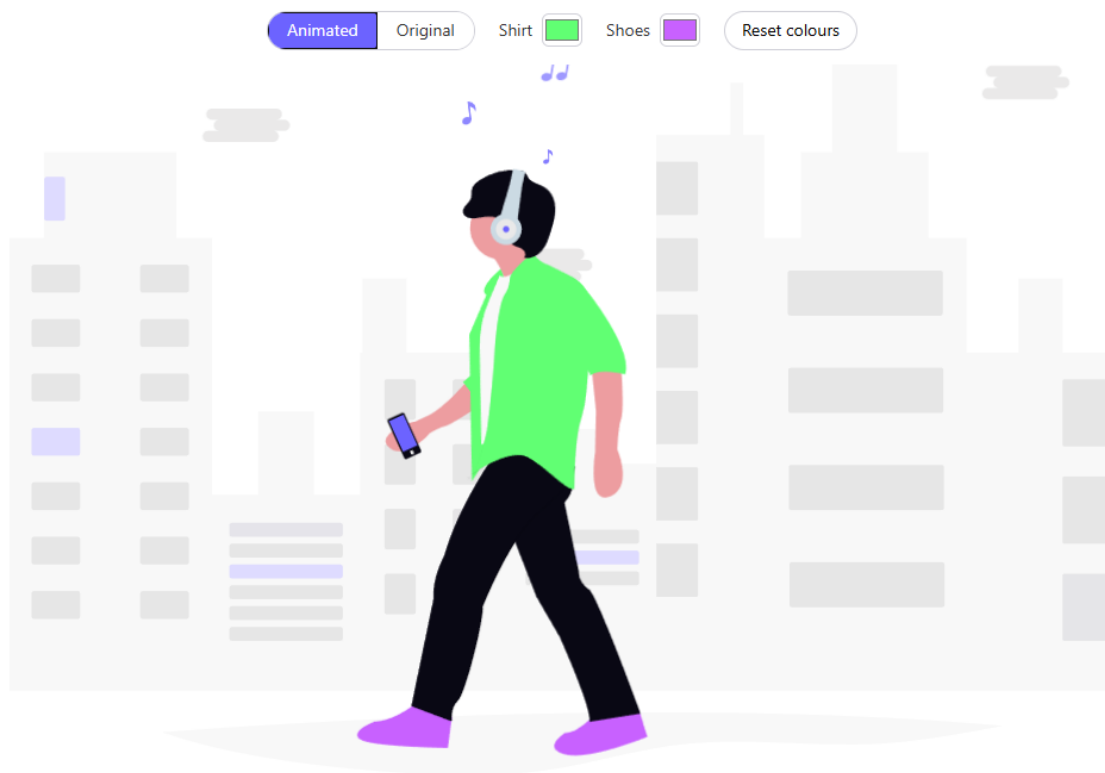
<https://www.adobe.com/in/creativecloud/file-types/image/vector/svg-file.html>

The SVG file format is a popular tool for displaying two-dimensional graphics, charts and illustrations on websites. Plus, as a vector file, it can be scaled up or down without losing any of its resolution.



Article – Animate Static SVG

<https://undraw.co/illustrations>



Article – Kanban

To do3

GovernanceHigh

Draft AI usage policy

DK3 subtasks · 1/9

Due 14 Oct

EnablementMedium

Prompt library for Finance

MJ2 subtasks · 1/6

Due 21 Oct

EngagementLow

Plan internal hackathon

KN2 subtasks · 1/6

Due 30 Oct

In progress3

RolloutHigh

Licence rollout, wave 2

AM3 subtasks · 7/10

Due 07 Oct

InsightsMedium

Readiness survey analysis

RG3 subtasks · 4/9

Due 10 Oct

PilotMedium

Sales email assistant pilot

DK2 subtasks · 4/6

Due 18 Oct

Review2

CommsLow

Monthly newsletter, issue 4

KN2 subtasks · 5/6

Due 25 Sep

InsightsHigh

Leadership dashboard

RG2 subtasks · 5/7

Due 28 Sep

Done2

EnablementMedium

Kick-off workshop

MJ2 subtasks · 6/6

Due 05 Sep

GovernanceHigh

Tool shortlist

DK2 subtasks · 6/6

Due 02 Sep

Click a card for task details, then click a subtask for its checklist and activity. Drag cards between columns to change status. Esc goes back.

Sample data

To do3

GovernanceHigh

Draft AI usage policy

DK3 subtasks · 1/9

Due 14 Oct

EnablementMedium

Prompt library for Finance

MJ2 subtasks · 1/6

Due 21 Oct

EngagementLow

Plan internal hackathon

KN2 subtasks · 1/6

Due 30 Oct

In progress3

RolloutHigh

Licence rollout, wave 2

AM3 subtasks · 7/10

Due 07 Oct

InsightsMedium

Readiness survey analysis

RG3 subtasks · 4/9

Due 10 Oct

PilotMedium

Sales email assistant pilot

DK2 subtasks · 4/6

Due 18 Oct

Review

Comms

Monthly newsletter, iss

KN2 subtasks · 5/6

Insights

Leadership dashboard

RG2 subtasks · 5/7

Board · Licence rollout, wave 2

Detail level 1

X

Rollout

Licence rollout, wave 2

Assign licences to the second group of 150 users and run their onboarding sessions.

Status

In progress

Owner

AMAsha Menon

Due

07 Oct 2026

Priority

High

PROGRESS

7 of 10 checklist items

70%

SUBTASKS

Click one for its checklist and activity

AM

Confirm user list

Asha Menon · 1 day · 3/3 done

Done

DK

Assign licences

Dev Kapoor · 1 day · 2/3 done

In progress

MJ

Run onboarding sessions

Meera Joshi · 4 days · 2/4 done

In progress

Click a card for task details, then click a subtask for its checklist and activity. Drag cards between columns to change status. Esc goes back.

To do3

GovernanceHigh

Draft AI usage policy

DK3 subtasks · 1/9

Due 14 Oct

EnablementMedium

Prompt library for Finance

MJ2 subtasks · 1/6

Due 21 Oct

EngagementLow

Plan internal hackathon

KN2 subtasks · 1/6

Due 30 Oct

In progress3

RolloutHigh

Licence rollout, wave 2

AM3 subtasks · 7/10

Due 07 Oct

InsightsMedium

Readiness survey analysis

RG3 subtasks · 4/9

Due 10 Oct

PilotMedium

Sales email assistant pilot

DK2 subtasks · 4/6

Due 18 Oct

Review

Comms

Monthly newsletter, iss

KN2 subtasks · 5/6

Insights

Leadership dashboard

RG2 subtasks · 5/7

Board > Licence rollout, wave 2 > Confirm user list

Detail level 2

×

Rollout

Confirm user list

Part of: Licence rollout, wave 2

Owner

AM

Asha Menon

Effort

1 day

Status

Done

Checklist

3 of 3

NOTES

Final list of wave 2 users signed off by each department head.

CHECKLIST

Ticking items updates the card on the board

✓

Get lists from departments

✓

Remove duplicates

✓

Sign-off from heads

ACTIVITY

Signed off

13 Sep 2026

Duplicates removed, 148 users

12 Sep 2026

Lists received from 6 departments

10 Sep 2026

Click a card for task details, then click a subtask for its checklist and activity. Drag cards between columns to change status. Esc goes back.

Article – 3D Models

<https://get.webgl.org/>

Crane motions

Drag to move the crane

Long travel

12.0 m

Cross travel

12.0 m

Hoist

6.4 m

Run lift cycle

Park the crane

View

Drag to orbit, scroll to zoom

Iso

Front

Side

Top

Exploded view

0%

Labels

Auto rotate

Wireframe

Shadows

Plant around it

Assemblies

Hover to highlight

Main girders

Two welded box girders, 1.8 m deep

Trolley, or crab

Rope drum, hoist motor, head sheaves

Hook block

Four falls of rope, ramshorn hook

Portal legs

Lattice braced, striped at the base

End carriages

Eight double flanged wheels

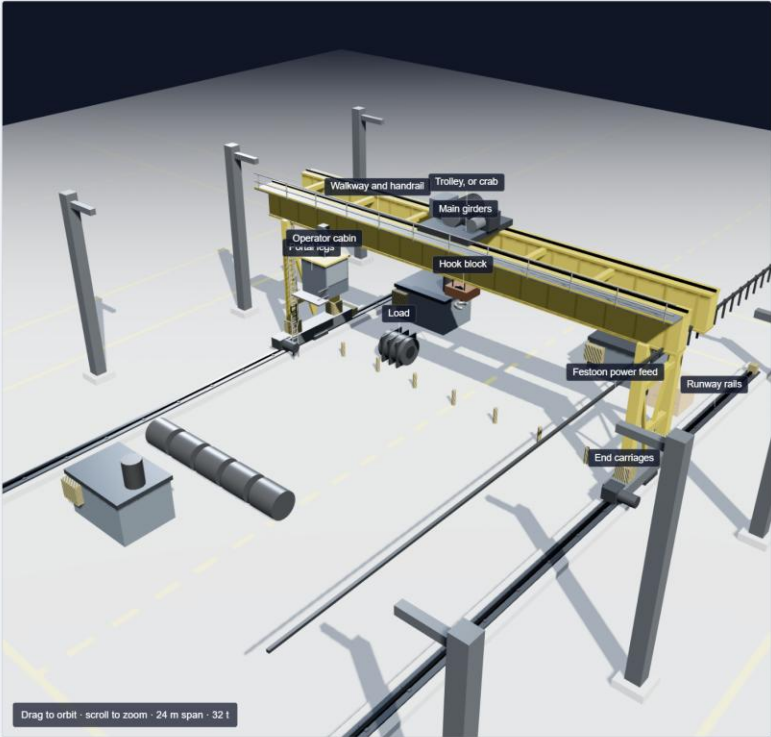
Runway rails

A75 rail on sole plates, with end stops

Walkway and handrail

Grating, toe board, twin rails

Operator cabin



Labels in the 3D model: Walkway and handrail, Trolley, or crab, Main girders, Operator cabin, Hook block, Load, Festoon power feed, Runway rails, End carriages.

Drag to orbit - scroll to zoom - 24 m span - 32 t

Design data

Type	Double girder goliath
Safe working load	32 t + 5 t aux
Span, rail to rail	24.0 m
Runway length	40.0 m
Height under girder	9.0 m
Lift height	8.2 m
Girder depth	1.80 m
Wheels per corner	2, double flanged
Rail section	A75 on sole plate
Long travel speed	32 m/min
Cross travel speed	20 m/min
Hoist speed	5 m/min, 0.8 creep
Duty group	FEM 2m / M5
Power feed	Festoon, 415 V
Control	Cabin and radio

Position

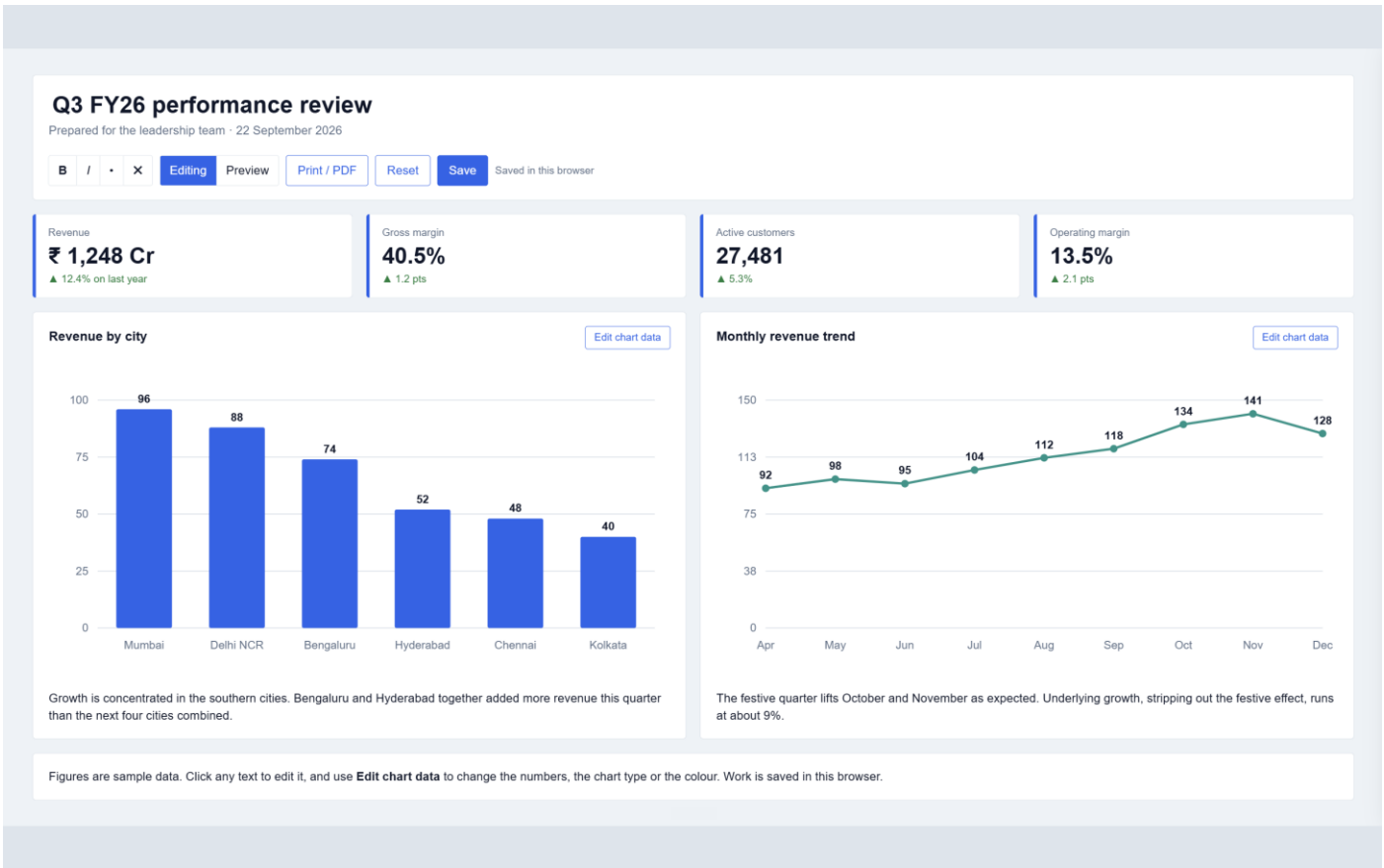
Live

Long travel	12.0 m
Cross travel	12.0 m
Hook height	6.39 m
Rope drawn	16.3 m
On the hook	empty

Notes

Dimensions are a worked example for a medium-duty plant crane, not a certified design. Swap the numbers at the top of the script and the model rebuilds to match.

Article – Editable Report



Article – Dynamic Dashboard connected to Google Sheets

Prompt:

Create code for google sheets

Sheet url is:

https://docs.google.com/spreadsheets/d/1wJMXo8_jN1BHdFBzTqi5bcNhosShjP9FPzsRUndbbWc/edit?gid=846480193#gid=846480193

It should make an html dashboard showing:

Column chart for manager wise average rating, KPIs for Sum of Sales, Average Rating and Average Sales

User

	A	B	C	D	E	F
1	Region	Item	Sales	Manager	Date	Rating
2	East	Laptop	\$12000	Sarah Lee	01/10/2025	4.5
3	West	Smartphone	\$8500	John Doe	02/10/2025	4.2
4	Central	Tablet	\$5500	Alex Kim	03/10/2025	3.9
5	South	Monitor	\$3200	Maria Garcia	04/10/2025	4.8
6	East	Keyboard	\$800	Sarah Lee	05/10/2025	4

create code for google sheets

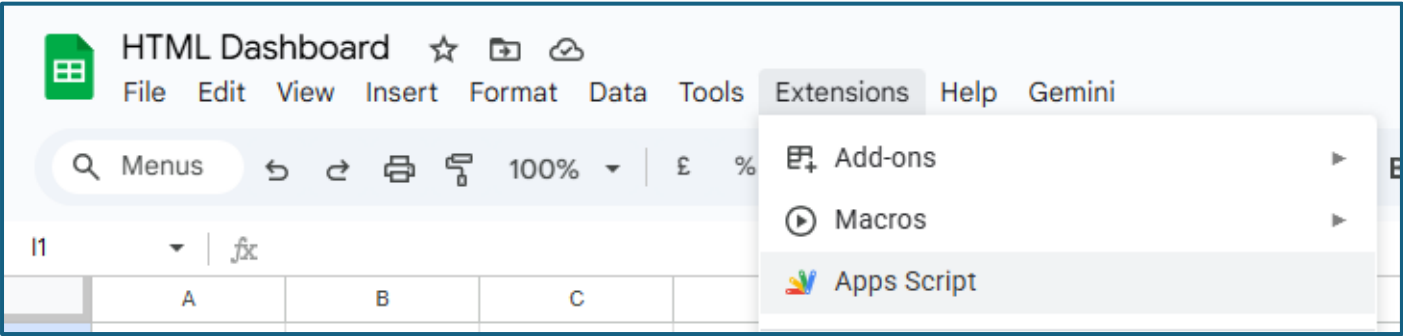
sheet url is:

https://docs.google.com/spreadsheets/d/1wJMXo8_jN1BHdFBzTqi5bcNhosShjP9FPzsRUndbbWc/edit?gid=846480193#gid=846480193

it should make an html dashboard showing:

column chart for manager wise average rating

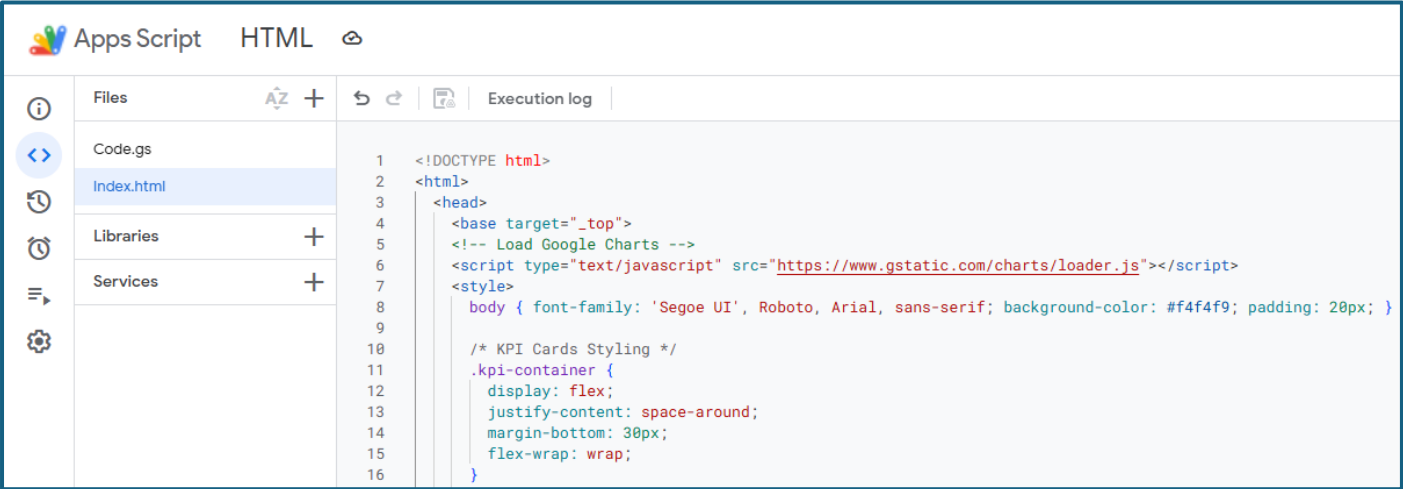
KPIs for Sum of Sales, Average Rating and Average Sales



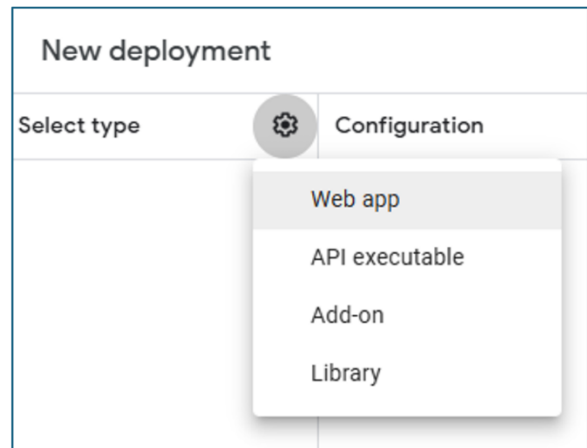
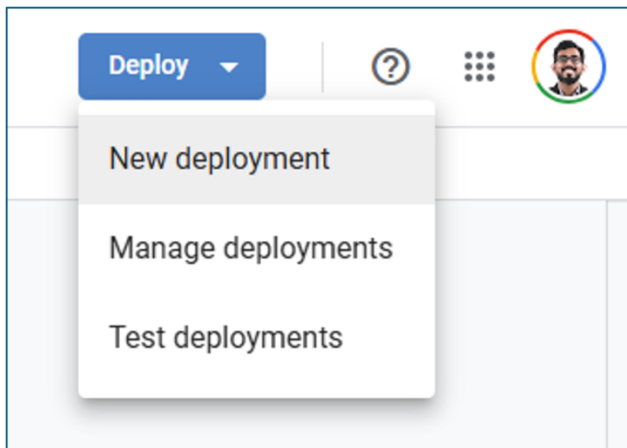
Code.gs



Index.html



Deploy > New deployment > Web app



New deployment

New deployment

Select type

Web app

Configuration

Description

New description

HTML Dashboard

Web app

Execute as

Me (havish.madhvapaty@gmail.com)

The web app will be authorised to run using your account data.

Who has access

Only myself

This can also be used as a library. [Learn more](#)

Cancel

Deploy

New deployment

Deployment successfully updated.

Version 1 on 11 Jan 2026, 20:02

Deployment ID

AKfycbzYMYuBUYbB2AV9FxtPk-3l2gdJ3ubW8wZup5QjOmm_4plvrHvhLRtXwIAAQqABXLyljQ

Copy

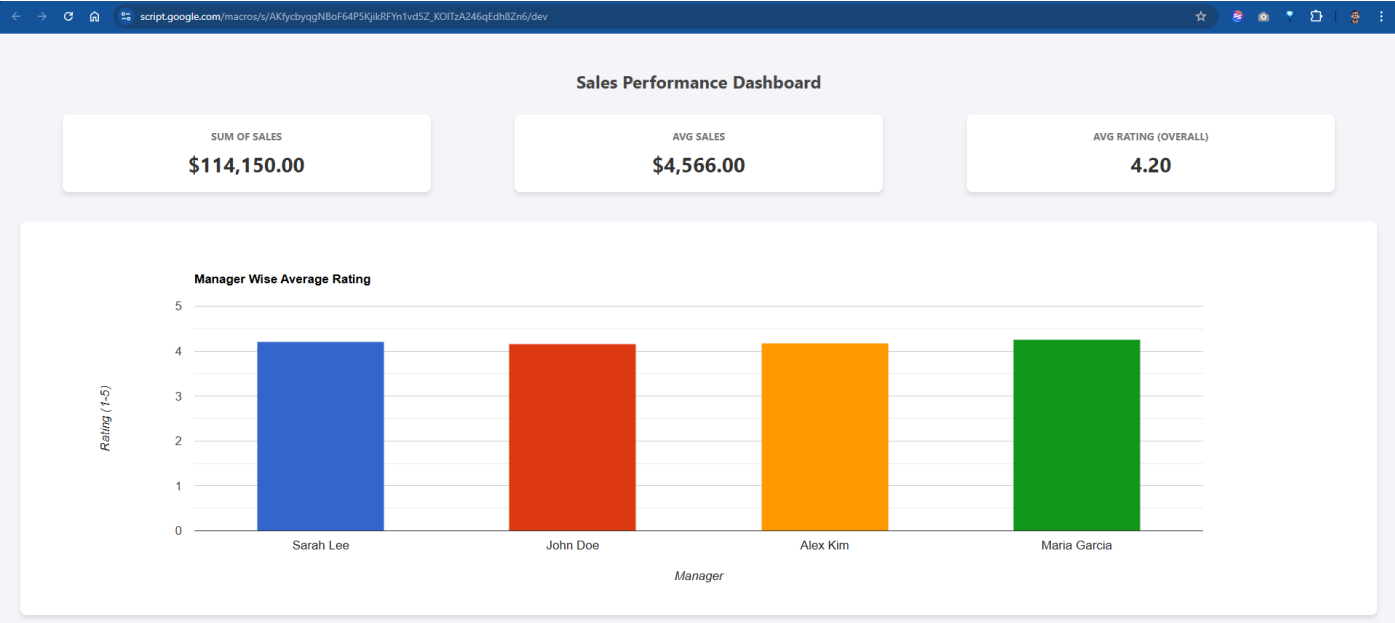
Web app

URL

https://script.google.com/macros/s/AKfycbzYMYuBUYbB2AV9FxtPk-3l2gdJ3ubW8wZup5QjOmm_4plvrHvhLRtXwIAAQq...

Copy

Final Dashboard



https://docs.google.com/spreadsheets/d/1wJMXo8_jN1BHdFBzTqi5bcNhosShjP9FPzsRUndbbWc/edit?gid=846480193#gid=846480193

https://script.google.com/macros/s/AKfycbyqgNBoF64P5KjikRFYn1vd5Z_KOITzA246qEdh8Zn6/dev

Part — Content and Interaction

Article – Interact

Ten ways a page can be interactive

Every demo below works. Nothing is loaded from the internet. Press `?` for the shortcuts, `/` to jump to the search box.

01 Tabs

JavaScript

One panel visible at a time. Click a tab, or focus one and use the arrow keys.

Overview Numbers Notes

Tabs save space when the content splits cleanly into a few views that are rarely needed at once.

02 Accordion

No JavaScript

Built on the native `<details>` element. The browser handles opening and closing.

What it is

Sections that expand one at a time, with no script at all.

When to use it

What to watch

03 Split pane

JavaScript

Drag the divider to resize. Double-click it to go back to an even split.

Left

Drag the grey bar. The two widths always add up to the full width, so nothing overflows.

Right

Pointer events make this work with a mouse, a trackpad and touch, all from the same code.

Split: 50 / 50

04 Keyboard shortcuts

JavaScript

Press any key to see it captured. `/` focuses search, `?` opens the shortcut sheet, `Esc` closes things.

Press `/` from anywhere to land here

Last key: `Cmd+Shift+Shift`

05 Modal and drawer

JavaScript

A dialog takes over the page; a drawer slides in beside it. Both close on `Esc` or a click outside.

Open modal

Open drawer

06 Tooltip and popover

Mostly CSS

A tooltip appears on hover and explains. A popover needs a click and stays until dismissed.

Gross margin was 38.2% this quarter.

How is this calculated?

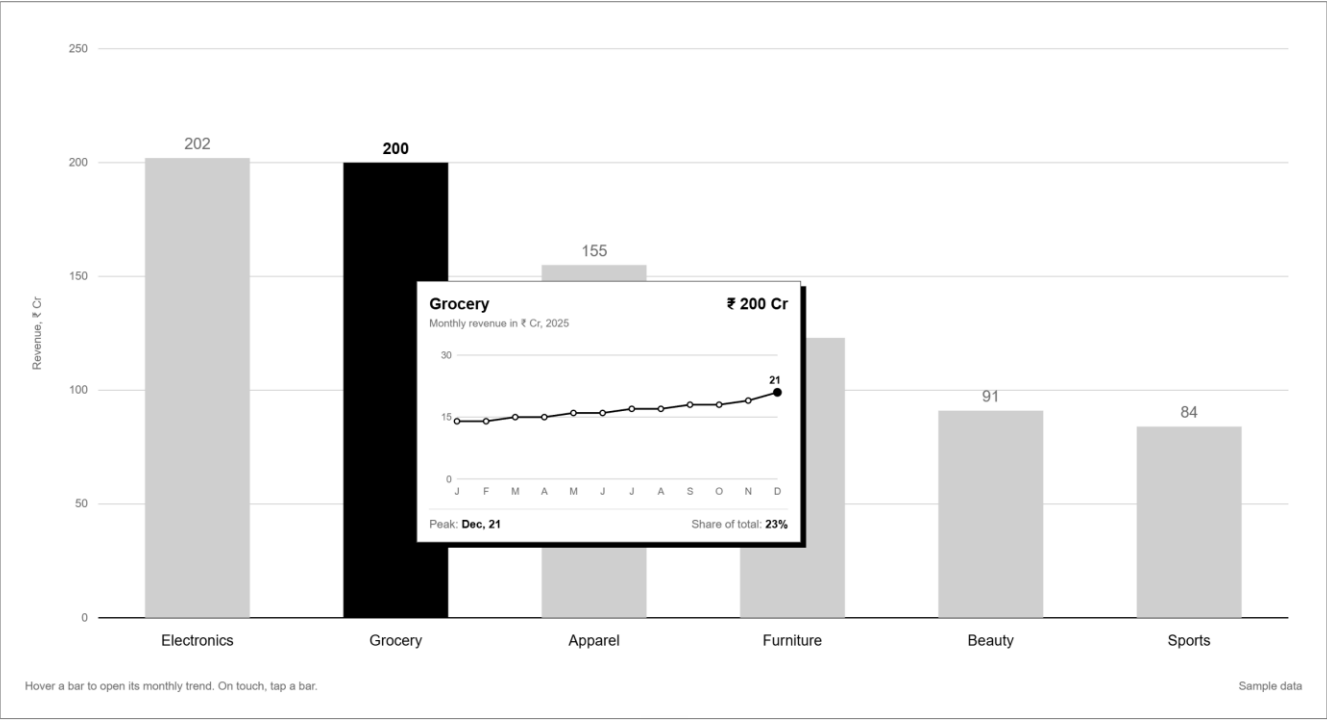
07 Drag and drop

JavaScript

08 Live filter and sort

JavaScript

Article – Tooltip



Article – MCQs

Results

Score, breakdown, and every question with its explanation

Question pool

What goes into the paper

TOPICS

StrategyAdoptionRiskTechniqueSample

DIFFICULTY

EasyMediumHard

Turn on adaptive below and the quiz picks the level for you, question by question.

QUESTION TYPES

Single answerMulti selectTrue or falseNumeric

HOW MANY QUESTIONS

10

4 questions match this selection, out of 36 in the bank. The quiz will use all 4.

PASTE YOUR OWN QUESTIONS

Strategy | 2 | Which metric best shows adoption has taken hold? | Weekly active users ; Licences issued ; Prompts run ; Training attendance | 1 | Weekly use is the habit measure; the others are inputs.

Add pasted questionsShow the format

Rules

How it is marked and paced

FEEDBACK

Instant feedbackExam mode

You see right or wrong, and the reason, straight after each question. You cannot go back.

MARK SCHEME

+1 / 0+4 / -1+1 / -0.5

TIMING

No timerPer questionWhole quiz

Questions are untimed. Good for learning, poor for exam practice.

Shuffle the questions

Shuffle the options

Adaptive difficulty

Ask for confidence

Partial credit on multi-select

Allow flag and revisit

What each setting does

Instant vs exam — whether the answer is revealed as you go or held until the end

Mark scheme — negative marking changes guessing behaviour more than any other setting

Per question vs whole quiz — a per-question clock auto-advances; a whole-quiz clock lets you budget your own time

Shuffle — options and questions are reordered so the answer is never in the same place twice

Adaptive — get it right and the next one is harder, get it wrong and it eases off

Confidence — flags the answers you were sure about and still got wrong

Partial credit — a multi-select scores proportionally rather than all or nothing

Flag and revisit — mark a question to come back to, and jump around from the palette

Start the quizQuick practiceExam simulation

Questions

4 of 4 answered

1234

AnsweredFlagged

Not seen

Finish and see results

Question 4 of 4

Hard

Sample

Multi select

Choose all that apply

Which of these are prime numbers? Choose all that apply.

19

217

32

421

Partly correct, +0.5 marks

2 and 17 have no divisors other than 1 and themselves. $9 = 3 \times 3$ and $21 = 3 \times 7$.

See results

Flag for review

Keys 1–6 pick an option, Enter submits

Confidence

Sure

Not sure

A confident wrong answer is the one worth studying.

Live

Answered4 of 4

Correct so far2

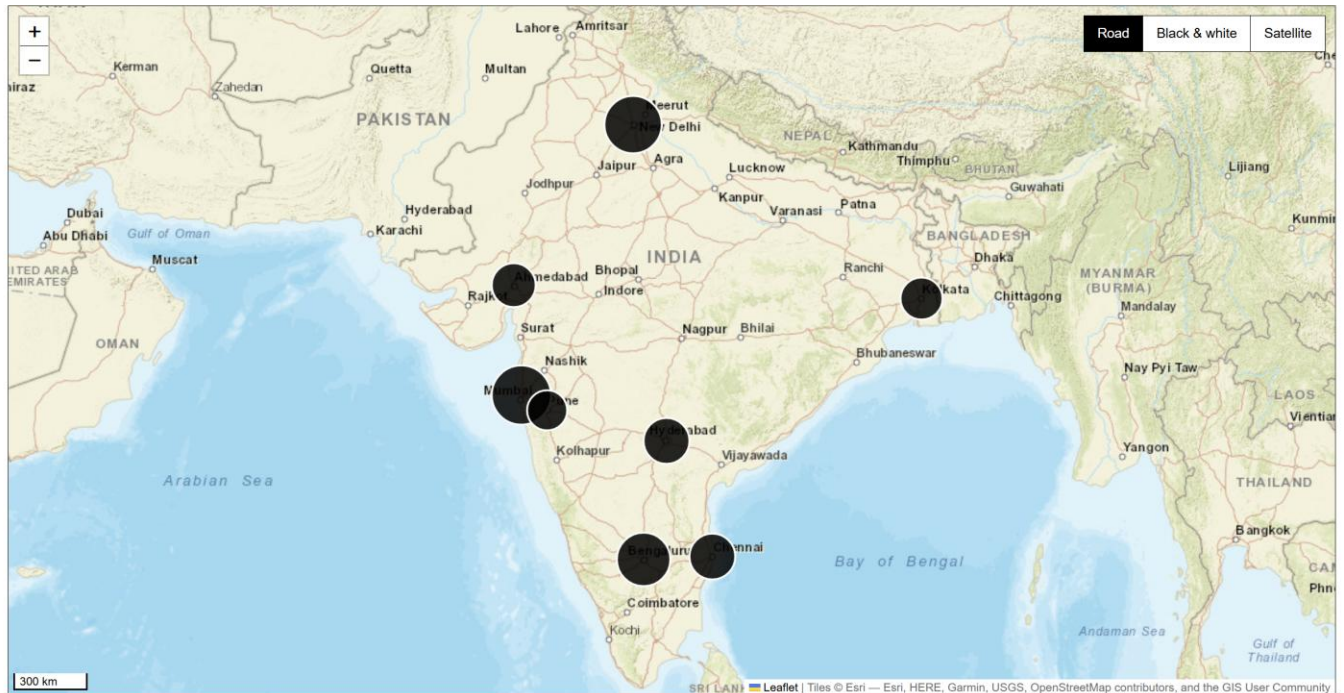
Marks2.5 / 4

Difficulty nowFixed

Elapsed25s

Article – Maps

<https://leafletjs.com/>



Hover a circle to see revenue by city. Circle size shows revenue. Use the switch on the map to change the map type.

Sample data

Article – Globe

Prompt:

Make an html animation that I can keep on my wix website that I manage.

I should be able to embed the code.

A globe that shows we work in India, UAE, US, UK.



Article – Embed HTML

Prompt:

Embed this in an HTML page

```
<iframe src="https://docs.google.com/forms/d/e/1FAIpQLSdH0WfZzrbAXCh8xuR7JVz-2NaJk9K4OZRaoY63xqrH3XDyKQ/viewform?embedded=true" width="640" height="1229"
frameborder="0" marginheight="0" marginwidth="0">Loading...</iframe>
```

Generative .

Questions Responses 1 Settings

1 response

Summary Question

dbansal9705@gmail.com 1 of 1

Responses cannot be edited

Generative AI for HR Leaders

* Indicates required question

Make a copy
Move to trash
Pre-fill form
Embed HTML
Print
Apps Script
Get add-ons
Unpublish form
Keyboard shortcuts

Premium Fuels
Experience smoother drives with Speed and Speed 97 premium petrol.

Bharatgas
Safe and reliable LPG cooking gas delivery for every Indian home.

MAK Lubricants
High-performance engine oils and lubricants for longer engine life.

Aviation Services
Fueling the aviation industry with global quality standards.

SmartFleet
Efficient fuel management solutions for transport fleets and logistics.

Generative AI for HR Leaders

havish@havishmconsulting.com [Switch account](#)

* Indicates required question

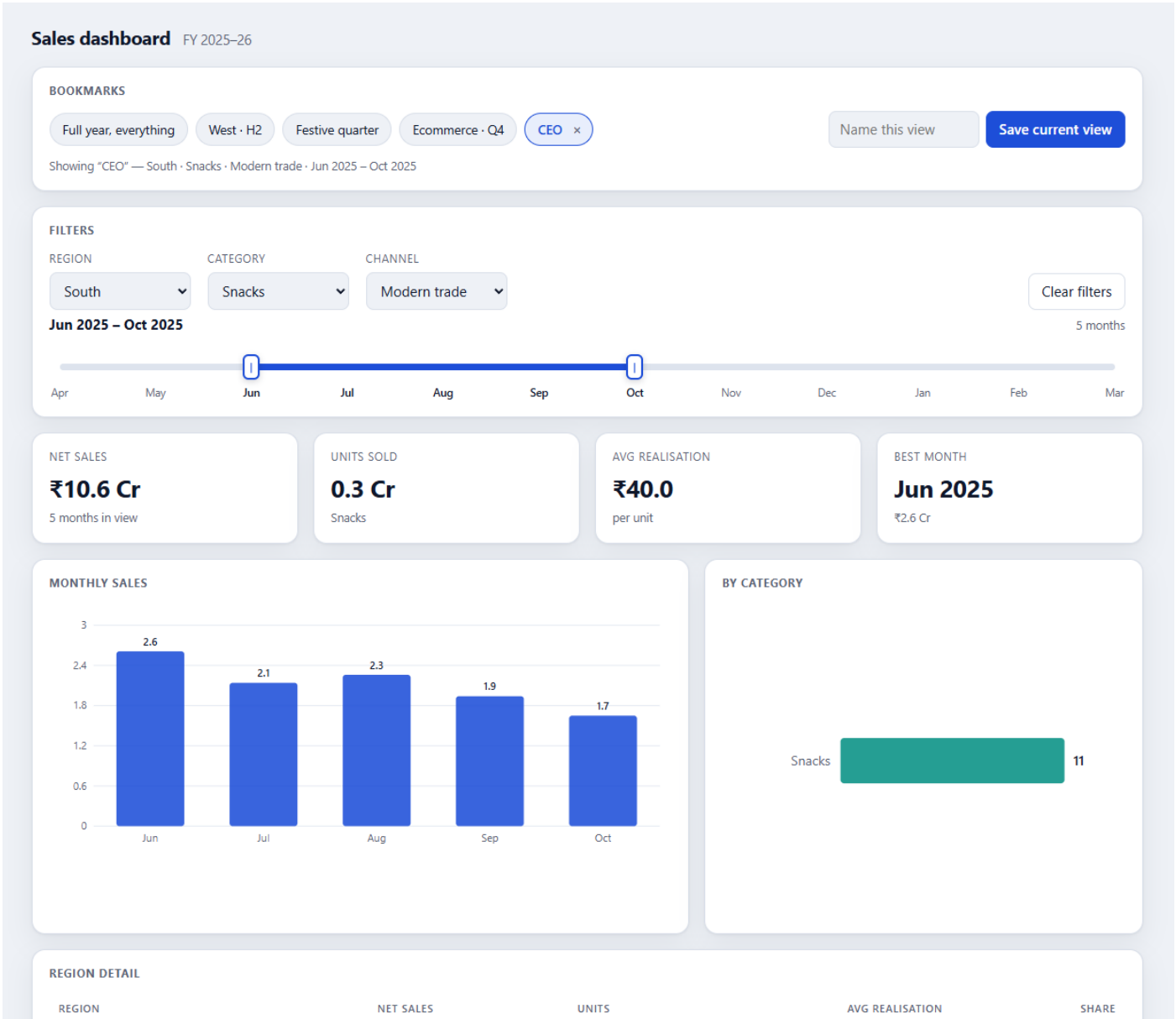
Email *

Your email

Full Name *

Your answer

Article – Bookmarks



Article – PDF Reader and Editor

PDF reader and editor IMK CHRO.pdf - 20 pages - 32.66 MB

Open a PDF

Undo

Save PDF

Read Edit pages

Select all Select none none selected

Delete selected pages

Extract selected pages

Merge another PDF



1 ☐ select



2 ☐ select



3 ☐ select



4 ☐ select



5 ☐ select



6 ☐ select



7 ☐ select



8 ☐ select



9 ☐ select



10 ☐ select



11 ☐ select



12 ☐ select



13 ☐ select



14 ☐ select



15 ☐ select



16 ☐ select



17 ☐ select



18 ☐ select



19 ☐ select



20 ☐ select

Part — Charts & Data Visualization

Article – Combine and Split

EXCEL UTILITY

Combine & Split Excel Files

// no server · runs entirely in your browser · supports .xlsx .xls .csv

⊕ Combine Files

🔗 Split File

UPLOAD FILES



[Click to browse](#) or drag & drop files here
Supports .xlsx · .xls · .csv – multiple files allowed

OPTIONS

SHEET FROM EACH FILE

First sheet only



OUTPUT FILENAME

combined_output

OUTPUT FORMAT

Excel (.xlsx)

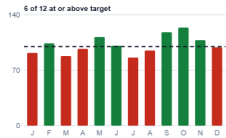


- ☒ Keep header row from first file only (remove duplicates)
- ☐ Add "Source File" column to track origin

⊕ COMBINE & DOWNLOAD

Article – Reference Lines

01 Above and below the line



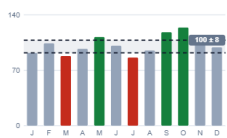
Your suggestion. One colour for months that beat the **target of 100**, another for those that miss. The eye sorts the year before it reads a single number.

02 Labelled line



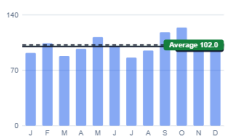
The plainest version. Bars stay one colour and the line carries a **badge** saying what it is, so nobody has to ask what the dashes mean.

03 Tolerance band



A band instead of a line, here **100 ± 8**. Inside the band is "on plan", so only the real exceptions take a colour.

04 Two lines: target and average



The target is the promise, the **average** is what actually happened. The gap between the two lines is the story.

05 Variance from the line



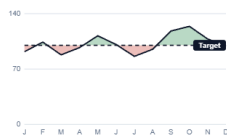
Drop the bars and plot only the **gap**. The line becomes the zero axis, which makes small misses visible at a glance.

06 Bullet chart



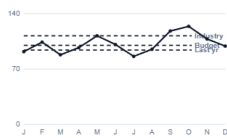
Shaded bands for poor, fair and good, with a **target tick**. Four KPIs fit where one gauge would have gone.

07 Shaded above and below



On a line chart, fill the area **between the line and the target** in two colours. Surplus and shortfall read as areas, not points.

08 Several benchmarks



Budget, last year and the industry figure together. Label each line **at its right end** rather than in a legend.

09 Pace line



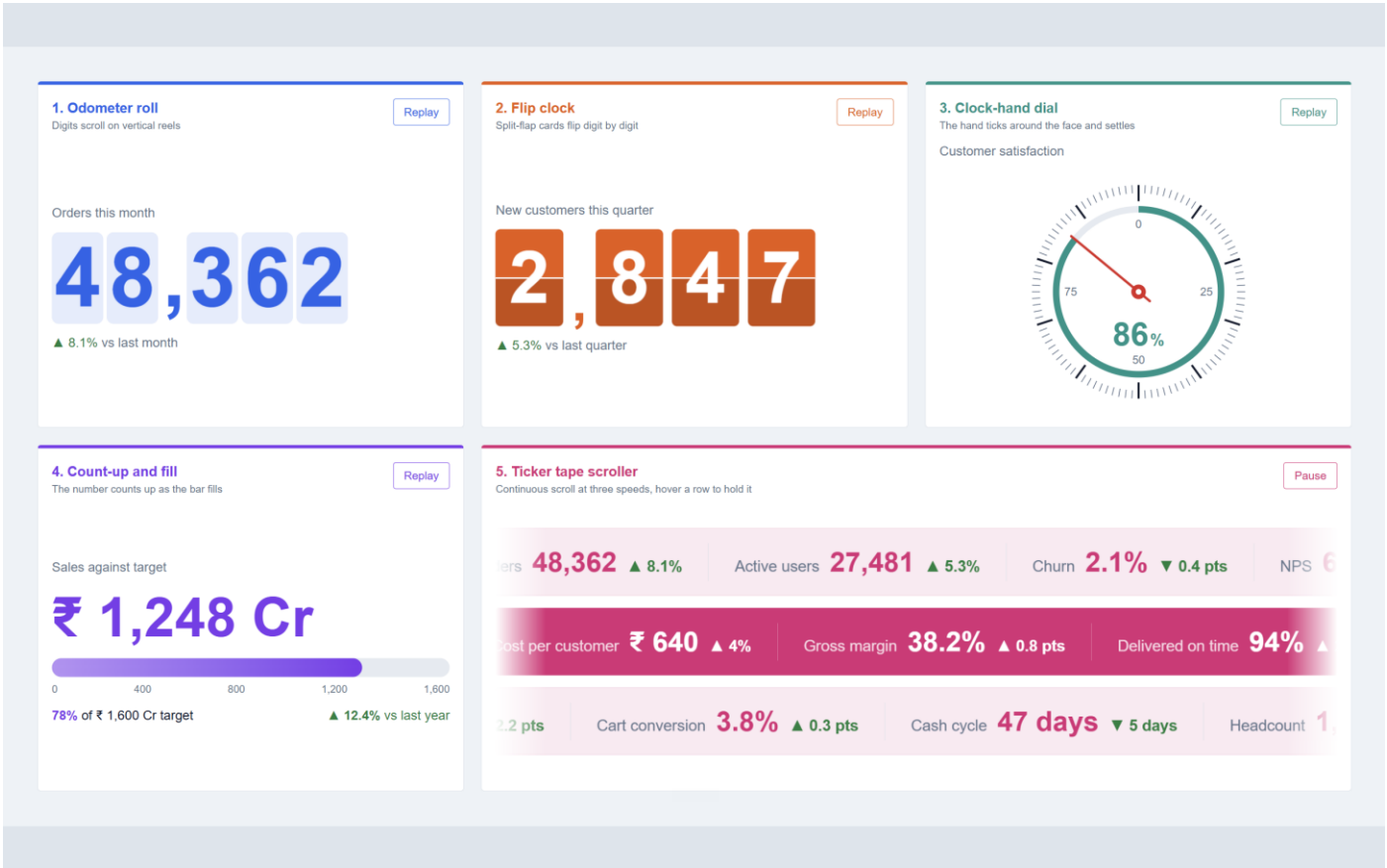
For anything cumulative the reference is a **ramp**, not a flat line. The shading shows whether you are ahead of pace or behind it.

10 Draggable threshold



Drag the line to ask "what if the target were higher?". At **101, 6 of 12** months clear it.

Article – Animated KPIs



4. Count-up and fill

The number counts up as the bar fills

Replay

Sales against target

₹ 1,248 Cr



78% of ₹ 1,600 Cr target

▲ 12.4% vs last year

5. Ticker tape scroller

Continuous scroll at three speeds, hover a row to hold it

Pause

Customers 48,362 ▲ 8.1%

Active users 27,481 ▲ 5.3%

Churn 2.1% ▼ 0.4 pts

NPS 6

Cost per customer ₹ 640 ▲ 4%

Gross margin 38.2% ▲ 0.8 pts

Delivered on time 94% ▲

2.2 pts

Cart conversion 3.8% ▲ 0.3 pts

Cash cycle 47 days ▼ 5 days

Headcount 1

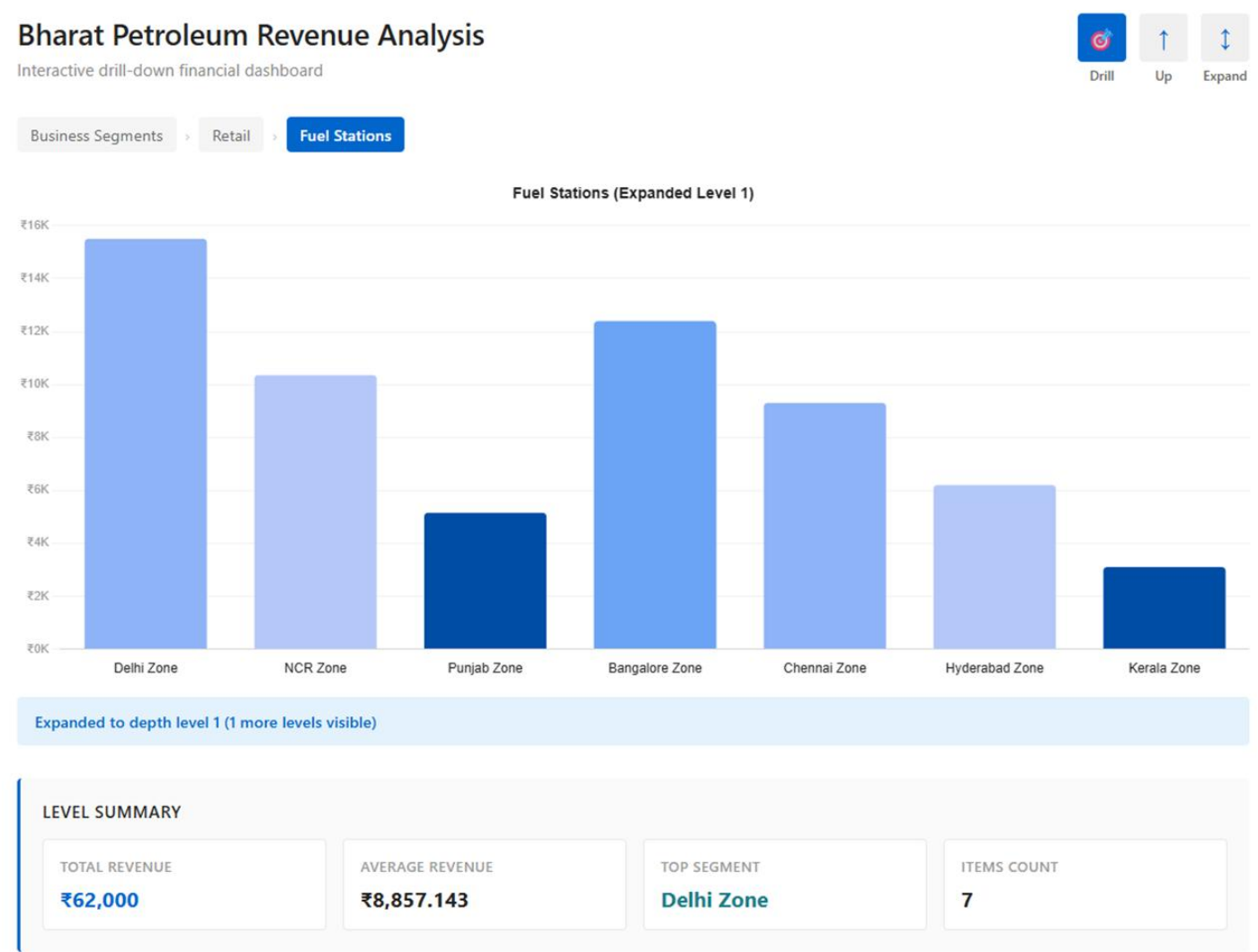
Article – Heatmap



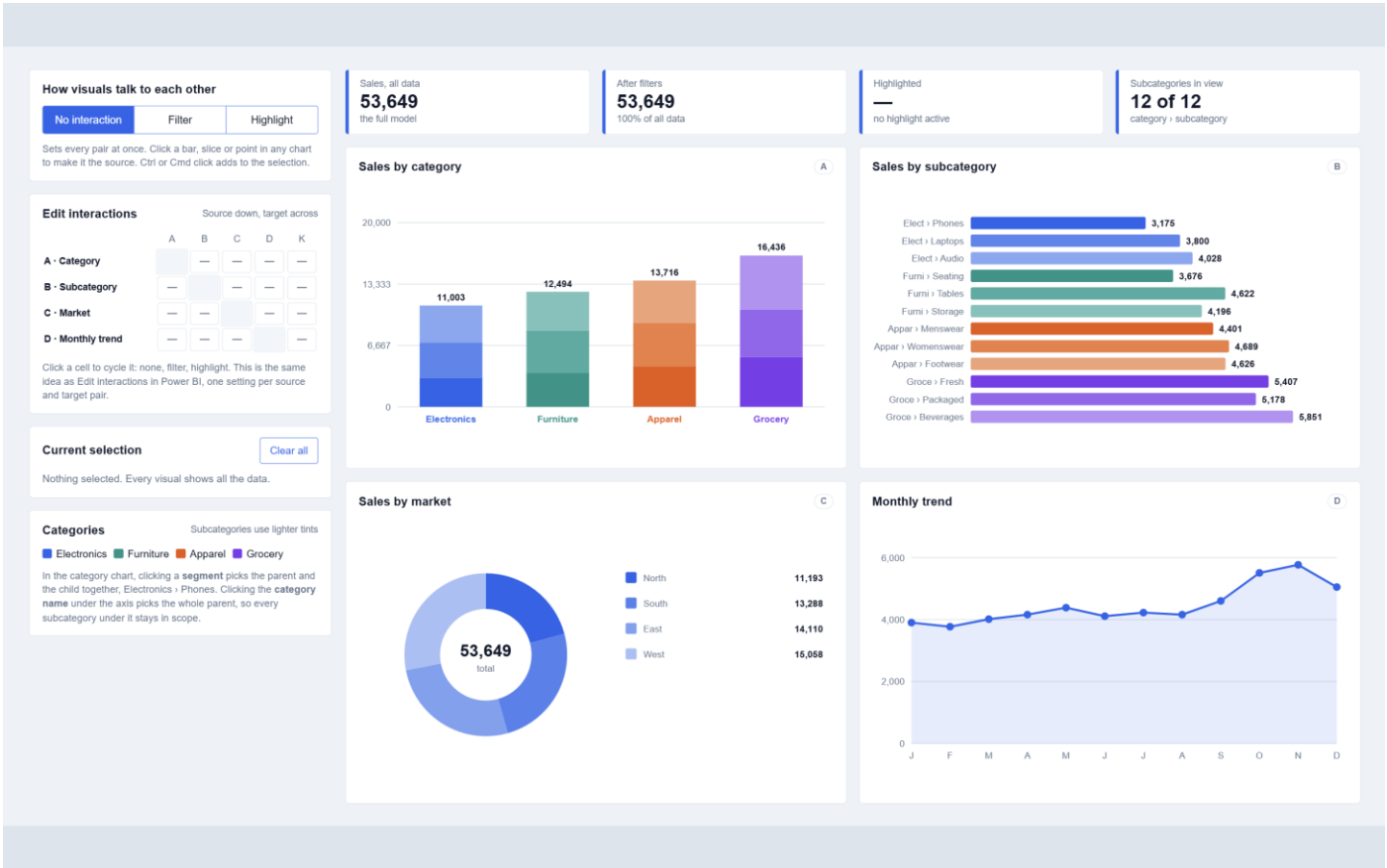
Article – Drill

Prompt:

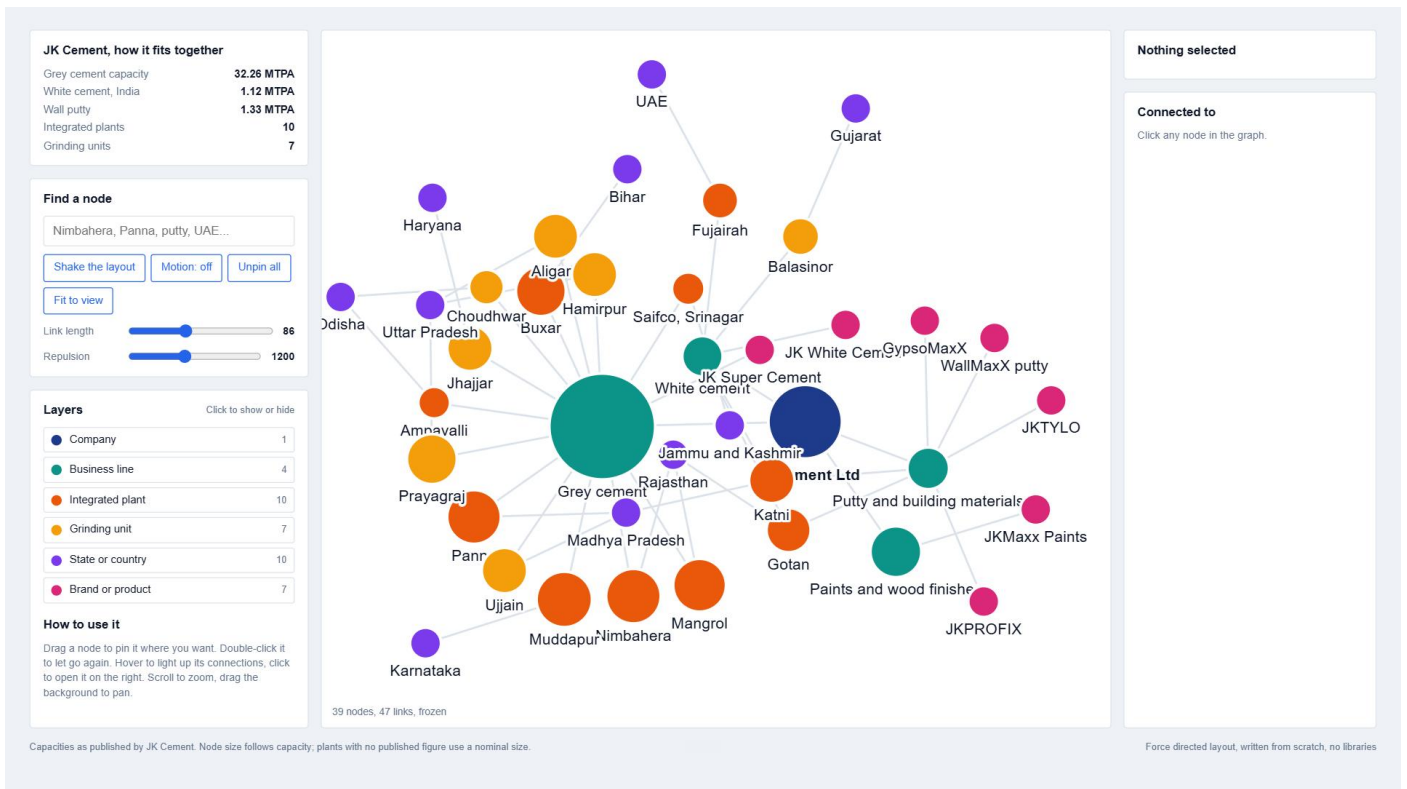
Create an interactive HTML + JavaScript column chart that supports drill-down (click a bar to expand into sub-categories) and drill-up (Back button). Include expand/collapse controls to show/hide deeper levels, with animated transitions and breadcrumb navigation.



Article – Filter and Highlight



Article – Force Directed Graph



Part — Media and Export

Article – Export Options

Most people who build something with AI stop one step too early. They get a working page — a dashboard, a tracker, a calculator — and then send a screenshot of it to their colleague. The page is good. The screenshot is not. It cannot be sorted, filtered, filed or forwarded, and the moment the underlying numbers change it becomes a lie.

The gap is not a technical one. A single HTML file, with nothing behind it, can hand the user an Excel workbook, a Word document, a PDF, a PowerPoint deck or a zip of all four. Very few people building with AI ask for this, because they do not know it is possible. It is worth knowing, because the export button is usually what turns a demo into a tool people actually use.

Why exports decide whether your tool survives

Every business tool sits inside a workflow that already exists. The finance analyst still has to paste the numbers into the month-end pack. The consultant still has to attach something to the email. The team lead still has to file the report.

If your page cannot produce the artefact that workflow expects, one of two things happens. Either the user retypes the output by hand, which they will resent and eventually stop doing, or they abandon the tool and go back to the spreadsheet. Neither failure looks like a technical failure. It looks like "we tried that AI thing and it did not stick".

An export button is therefore not a nice finishing touch. It is the join between your tool and the organisation around it.

The two tiers of export

Everything a browser can produce falls into one of two tiers, and the difference matters when you are specifying what you want.

Tier one needs nothing at all. Plain browser code can write a file and hand it to the user. This covers CSV, TSV, JSON, Markdown, HTML, images in PNG, JPEG or SVG, and a rich clipboard copy that pastes into Word or Excel with its formatting intact. It also covers PDF, through the browser's own print dialog and its Save as PDF option, which produces the sharpest text of any method available.

Exhibit: Formats a browser can produce on its own

Format	What it gives you	Worth knowing
CSV	The data, ready for any other system	Add a byte order mark or Excel mangles ₹ and accented names
TSV	Tab-separated data	Pastes into a spreadsheet cleanly

JSON	Structured data for another tool or script	For handing to a developer, not to a colleague
Markdown	A table for notes, wikis and documentation	Plain text, survives anywhere
HTML	A standalone page anyone can open	No software needed at the other end
Excel (.xls)	A formatted sheet, via an HTML table saved with an Excel file type	Excel shows a one-time format warning. Fine internally, not for clients
Word (.doc)	Headings and tables, via the same trick	Same warning, same rule
Selectable-text PDF	The browser's print dialog, Save as PDF	Sharpest text of any method, and free
SVG	A chart as true vector	Opens in Illustrator, Figma and PowerPoint, stays sharp at any size
PNG / JPEG	A chart as an image	Render at 2x so it holds up on a high-resolution screen
Clipboard, rich text	A table that pastes into Word or Excel with formatting intact	The most-used export in practice, and the most overlooked
Clipboard, image	A chart that pastes straight into a slide	Saves the save-then-insert round trip
.ics / .vcf	Calendar invite, contact card	Same technique, different text format

There are two useful tricks in this tier. A table written as HTML but saved with an Excel file type opens in Excel as a formatted sheet, and the same trick with a Word file type opens in Word with headings and tables intact. Both show a one-time warning that the file format does not match the extension. For a quick internal deliverable that warning is a fair trade. For anything going to a client, it is not.

Tier two needs a library. A genuine .xlsx with multiple sheets and no warning, a genuine .docx, a PDF you lay out yourself, an editable PowerPoint deck, a zip bundle of several files at once. These need a code library, typically half a megabyte to a megabyte each.

Exhibit: Formats that need a library embedded in the file

Format	Library	Approximate size	What it adds over the free version
Excel (.xlsx)	SheetJS	900 KB	A genuine workbook, multiple sheets, no warning dialog

Excel with full styling	ExcelJS	1 MB	Colours, borders, column widths, frozen panes
Word (.docx)	docx.js	700 KB	A genuine Word document with proper headings, tables and page breaks
PDF, vector	jsPDF with AutoTable	400 KB	You control the layout, text stays selectable, long tables split across pages
PDF, screenshot	html2canvas with jsPDF	600 KB	Pixel-identical to the screen, but the text becomes an image
PowerPoint (.pptx)	PptxGenJS	500 KB	Editable slides with real tables, not pictures of tables
Zip bundle	JSZip	100 KB	One download instead of five

The important point is that the library can be embedded inside the HTML file itself. Ask for that explicitly, because the default behaviour is to load libraries from the internet, and the day someone opens your file on a locked-down corporate laptop or on a plane, the page breaks. A self-contained file is larger and completely reliable. In a business setting, take the trade.

Which format for which job

The instinct is to offer every format. Resist it. Each format carries an implicit instruction about what the recipient should do next, and offering all of them signals that you did not think about it.

Exhibit: Matching the format to the situation

The situation	The format	Because
The data is going into another system or model	CSV	The most useful format on the list, and the most commonly forgotten
The recipient will re-sort, extend or recut the numbers	Excel	Put the summary on its own sheet
The deliverable is an argument, and someone will edit the wording	Word	Structure matters more than the data
The content is final and should not change	Selectable-text PDF	The default for anything leaving the organisation
The material will be presented	PowerPoint	An editable table beats a picture of a table
One chart needs to travel into someone else's deck	SVG, or PNG at 2x	The vector version stays sharp when resized

The recipient needs several of the above	Zip	One click instead of five
The output is going into an email	Rich-text clipboard copy	The most common action of all, and the biggest time saver

Two limits worth being clear about

The first is deliberate: everything described here happens on the user's own machine. No data is uploaded, nothing passes through a server. When the content is client data or salary figures or anything under an NDA, that is not a limitation, it is the main argument in favour of the approach.

The second is real: a browser file cannot save to a shared drive, post to a team channel or send an email on its own. Those require credentials, a connection and a system to connect to. The honest workaround is to export the file and let the person put it where it belongs. If a tool genuinely needs to write into a live system, it has outgrown a single HTML file, and that is a different conversation.

How to ask for it

The pattern is short, and it works whichever AI tool you use:

Add an export bar to this page. Excel with the data on one sheet and the summary on another, a PDF with the chart and the table, and a CSV. Embed the libraries in the file so it works with no internet. Show the filename after each download.

Three things in that instruction do the work. Naming the formats, rather than asking for exports in general. Saying what goes into each one, so the Excel file is not just the screen dumped into a grid. And requiring that it works offline, which is the single request most people leave out and most often regret.

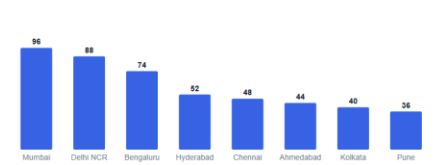
The takeaway

The value of a tool is not the analysis it performs. It is the artefact somebody else can act on. A page that produces a clean Excel file and a clean PDF will be used long after a more sophisticated page that only produces a view on screen.

Build the export before you polish the interface. It is the part that determines whether the work leaves your laptop.

Q3 FY26 revenue by city

₹ Cr · sample data



CITY	REGION	UNITS	REVENUE	MARGIN %	GROWTH %
Mumbai	West	18,240	96	38.2	+12.4
Delhi NCR	North	16,980	88	36.5	+9.8
Bengaluru	South	14,120	74	41	+18.6
Hyderabad	South	10,450	52	39.4	+15.2
Chennai	South	9,380	48	35.1	+6.4
Ahmedabad	West	8,610	44	34.8	+11.1
Kolkata	East	7,920	40	32.6	-2.3
Pune	West	7,140	36	37.9	+8.7
Total		92,840	478		

No library needed

Plain JavaScript

CSV

CSV file

Comma separated, with a BOM so Excel reads ₹ correctly

↓

TSV

Tab separated

Pastes straight into a spreadsheet

↓

JSON

JSON data

For feeding another tool or script

↓

MD

Markdown table

For notes, wikis and README files

↓

HTML

HTML file

A standalone page anyone can open

↓

XLS

Excel, legacy trick

HTML table with an Excel MIME type. Excel warns once on open

↓

DOC

Word, legacy trick

Same idea for Word. Keeps headings and tables

↓

PDF

PDF via print dialog

Choose Save as PDF. Sharpest text, no library

↓

SVG

Chart as SVG

True vector, opens in Illustrator or PowerPoint

↓

PNG

Chart as PNG

Rendered at 2x through a canvas

↓

JPEG

Chart as JPEG

Smaller file, no transparency

↓

COPY

Copy table as rich text

Paste into Word or Excel with formatting intact

↓

COPY

Copy chart as image

Paste straight into a slide

↓

Library embedded in this file

Still works offline

XLSX

NEW! Excel! No library

SheetJS - two sheets, no warning dialog

↓

DOCX

Real Word document

docx.js - headings and a proper table

↓

PDF

PDF, vector

jsPDF and autoTable - text stays selectable

↓

PDF

PDF, screenshot of the page

html2canvas - pixel perfect, text becomes an image

↓

PPTX

PowerPoint deck

PptxGenJS - editable table plus the chart

↓

ZIP

Everything in one zip

JSZip - CSV, JSON, chart PNG and the PDF

↓

Exports from this session

2 files

revenue.pdf	13.4 KB
revenue.pptx	176.4 KB

Article – Record Video

Record and edit video, entirely in the browser

Nothing is uploaded. The recording never leaves this machine.

Edit an existing file

1. Record

Choose a source, then press Start

No camera or screen connected yet.
Pick a source on the right and press Connect.

Start recording

Pause

Stop

Capture a still

Idle

Source

Camera

Screen

Screen + camera

CAMERA

Camera 1

MICROPHONE

Microphone 1

RESOLUTION

720p

FRAME RATE

30

COUNTDOWN

3 seconds

STOP AFTER

No limit

Mirror the camera

Record microphone

MICROPHONE LEVEL

Connect

Disconnect

Microphone

off

Recording as

video/mp4

2. Edit

Record something, or load a file, to start editing

Adjustments

Applied when you export

SPEED

0.5x

1x

1.5x

2x

CROP

Aspect

16:9

Source

0:16

Article – Browser Cache

What the Browser Keeps: Cache and Storage in a Single HTML File

There is a moment that happens to almost everyone who starts building with AI. You change something in your file, save it, refresh the browser, and nothing changes. The old version is still on screen. You change it again. Still nothing. After ten minutes of this you conclude the AI broke something.

It did not. The browser is showing you a copy it kept from earlier. Understanding what the browser keeps, and why, is one of those small pieces of knowledge that saves hours of confusion and quietly makes your tools better.

There are two separate things at work, and they get confused constantly. The first is the cache, which is how the browser stores the file itself. The second is storage, which is how your page remembers what the user did. Different mechanisms, different rules, different problems.

Part one: the cache, or why your changes do not appear

When a browser loads anything — a page, an image, a code library — it keeps a copy on disk. The next time that same thing is needed it uses the copy rather than fetching it again. This is why the second visit to a website feels instant.

The rule is that the browser decides how long to trust its copy, based on instructions from the server. For a file you are editing on your own machine, that logic still applies, and it is usually the reason your edits seem to vanish.

The fix takes two seconds once you know it: a hard refresh. **Ctrl+Shift+R** on Windows, **Cmd+Shift+R** on a Mac. That tells the browser to discard what it has and fetch everything again. If you are going to be iterating on a file for an hour, open the developer tools with F12 and tick "Disable cache" in the Network tab, which keeps the browser honest for as long as the tools are open.

Worth knowing: this is also why a page that loads a library from the internet works fine all week and then breaks on a plane. The library was never in your file. It was in the browser's cache, and the cache expired.

Part two: storage, or how your page remembers things

The cache is about the file. Storage is about the contents — the numbers someone typed, the tab they had open, the draft they had not finished. A single HTML file has four places to keep them, and choosing the wrong one is the most common mistake in this area.

Exhibit: The four places a page can keep data

Where	Survives a reload	Survives closing the tab	Shared between tabs	Rough limit	Use it for
A variable in the code	No	No	No	Whatever memory allows	Anything you are happy to lose
sessionStorage	Yes	No	No	About 5 MB	One sitting's work: the open tab, a filter, a step in a form
localStorage	Yes	Yes	Yes	About 5 MB	Preferences, drafts, the last version of a report
IndexedDB	Yes	Yes	Yes	Hundreds of MB	Large datasets, files, images, anything past a few MB

Two of these deserve a comment.

localStorage is the default choice and the one you will use ninety per cent of the time. It holds text, it is simple, and it survives a browser restart. Its limit is around five megabytes, which is far more than it sounds — a few thousand rows of a table fit comfortably.

IndexedDB is what you move to when you outgrow that: a year of daily records, uploaded images, a cached dataset. It is more work to use, which is why it is worth naming explicitly when you ask for it, rather than hoping it appears.

The boundary nobody explains

Storage is tied to the origin, which means roughly the website address the page came from. Data saved by a page on one address cannot be read by a page on another. This has three consequences that surprise people:

- Data does not travel with the file. If you email your report tool to a colleague, they receive the tool, not your saved work. Their browser starts empty.
- The same file in two places is two different stores. Open it from your Downloads folder, then open it from a shared drive, and the second copy will not see what the first one saved.
- Clearing browsing data clears it. If the user tidies up their browser, or opens the page in a private window, everything the page remembered is gone.

None of this makes storage a bad idea. It makes it the wrong place for anything that matters. The rule is simple: **storage is a convenience, not a record**. Anything the user would be upset to lose must be

exportable to a file. This is why the save button and the export button are two different things and you need both.

What this means in practice

Exhibit: Matching the mechanism to the need

What you want	Use	Why
Remember the user's preferences and filters	localStorage	Small, simple, survives a restart
Keep a draft while they work	localStorage, saved as they type	Costs almost nothing, prevents the worst outcome
Remember which tab or step they were on	sessionStorage	Should not follow them into tomorrow
Hold a large dataset the page loaded	IndexedDB	Past five megabytes, localStorage fails
Hold anything the user cannot afford to lose	An exported file	Storage is not a record and never was
Make your page work offline for good	Embed the libraries in the file	The cache is not a guarantee, only a convenience

How to ask for it

The instruction is short:

Save the user's edits to localStorage as they type, and reload them when the page opens. Add a Reset button. Tell the user plainly that the data is saved in this browser only and add an export to file.

The last clause is the one people leave out, and it is the one that prevents the complaint that always follows: *I lost everything*.

The takeaway

The cache is the browser being efficient with the file. Storage is your page being useful with the data. Both are working exactly as designed, and both look like bugs the first time they surprise you.

Two habits cover almost every case. Hard refresh while you build, so you are always looking at what you actually wrote. And treat anything the browser remembers as a courtesy to the user, never as the place their work lives.

Variable in the code

The simplest store, and the one people use by accident

Survives a reload

No

Survives closing the tab

No

Shared between tabs

No

Rough limit

Memory

0

page loads recorded here

Empty. Nothing survived, or nothing was saved yet.

Type something to save

Save

Read

Clear

sessionStorage

One sitting's work, in this tab only

Survives a reload

Yes

Survives closing the tab

No

Shared between tabs

No

Rough limit

~5 MB

0

page loads recorded here

Empty. Nothing survived, or nothing was saved yet.

Type something to save

Save

Read

Clear

localStorage

The default choice for preferences and drafts

Survives a reload

Yes

Survives closing the tab

Yes

Shared between tabs

Yes

Rough limit

~5 MB

7

page loads recorded here

hello

saved 23/09/2026, 23:04:43

74 B of text

Type something to save

Save

Read

Clear

IndexedDB

For datasets and files too big for the rest

Survives a reload

Yes

Survives closing the tab

Yes

Shared between tabs

Yes

Rough limit

Hundreds of MB

7

page loads recorded here

hello

saved 23/09/2026, 23:04:45

74 B of text

Type something to save

Save

Read

Clear

What happened

Newest first

23:10:31

This origin may store up to about 10240.00 MB in total, with 5.1 KB used.

23:10:31

IndexedDB: survived, 7 loads recorded.

23:10:31

localStorage: survived, 7 loads recorded.

23:10:31

Page opened. Opened as a local file.

Two habits worth keeping

• While you build:

hard refresh with `Ctrl + Shift + R`, or `Cmd + Shift + R` on a Mac, so you see the file you actually saved.

• When you ship:

treat everything above as a convenience. Storage belongs to this browser on this machine, it does not travel with the file, and a private window or a cleared history wipes it. Anything the user cannot afford to lose needs an export to a file.

Part — Branding and Visual Style

Article – Brand Colors

<https://coolors.co/>

coolors

 Wegic Modern, dynamic and visually stunning websites created from a simple business description. [Try Now](#)

Tools  [Go Pro](#) [Sign in](#) [Sign up](#)

The super fast
color palettes
generator!

AND MUCH MORE

Create perfect palettes for your designs or get inspired by thousands of beautiful color schemes.

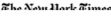
Start the Generator

Explore 10M+ Palettes



TRUSTED BY 8+ MILLION CREATIVE MINDS AND TOP COMPANIES

 slack   EA  DREAMWORKS   UBISOFT  Disney  airbnb  Dropbox  Microsoft  NETFLIX

 The New York Times  CNN  BBC

 Bluehost
Blog, website, or online store, we've got a plan that fits perfectly.
[Claim Offer](#)



Article – Color Picker

Image

Upload imageSample imagePick from screen

#4E4E4E · Dark grey · click to select



Grouped colours

Groups6Regroup

#BFB4AC21.4% of the image · orange

#1A191A20.1% of the image · Dark grey

#9D8D8618.8% of the image · orange

#E7E0DD15.1% of the image · Light orange

#48383413.0% of the image · Deep red

#83655411.6% of the image · orange

Selected colour

Save colour

#B7B6B6Light grey

HEX, click to copy#B7B6B6

RGB, click to copyrgb(183, 182, 182)

HSL, click to copyhsl(0, 1%, 72%)

CSS variable, click to copy--brand: #b7b6b6;

H0°

S1%

L72%

Tints and shades

Click any swatch to select it

Base

Comp

Analog -30

Analog +30

Triad 120

Triad 240

Split 150

Harmonies

Built from the selected colour

Base

Comp

Analog -30

Analog +30

Triad 120

Triad 240

Split 150

Contrast

WCAG ratio against white and black

2.02:1on white · text Fails · large Fails

10.38:1on black · text AAA · large AAA

Saved colours

Clear

Export

Copy hex listCopy CSS variablesDownload JSONDownload palette PNG

Part — Vibe Coding Platforms

<https://replit.com>

<https://lovable.dev>

<https://bolt.new>

<https://app.emergent.sh>

Gallery



IIM KOZHIKODE



NUS BUSINESS SCHOOL - SINGAPORE



SP JAIN DUBAI



MASTERS' UNION



BHARAT PETROLEUM – KOCHI REFINERY



NUVOCO CEMENT



JINDAL INDIA



BAHRI MAZROEI TRADING COMPANY – DUBAI



LANDMARK GROUP – DUBAI



AMBER ENTERPRISES



HMC ONE

AI STRATEGY. **BUILT TO SCALE.**

youtube.com/c/havishmadhvapaty

havishmconsulting.com