Use GenAI to make learning resources you can keep: workshop worksheet

A GenAI workshop for the Department of Management, LSE. This is the accessible text version of the interactive worksheet. Screens 1 to 8 are the main session; screens 9 to 13 are optional Go further screens, and screen 14 is troubleshooting help.

Contents

Main session

- Screen 1: Use GenAI to make learning resources you can keep, page 2
- Screen 2: Prepare your materials, page 3
- Screen 3: Set up your Project, page 4
- Screen 4: Your turn: choose your activity, page 6
- Screen 5: Activity A: A revision activity, page 6
- Screen 6: Activity B: A map of your course, page 8
- Screen 7: Activity C: An animated explainer video, page 9
- Screen 8: Skills to take with you, page 11

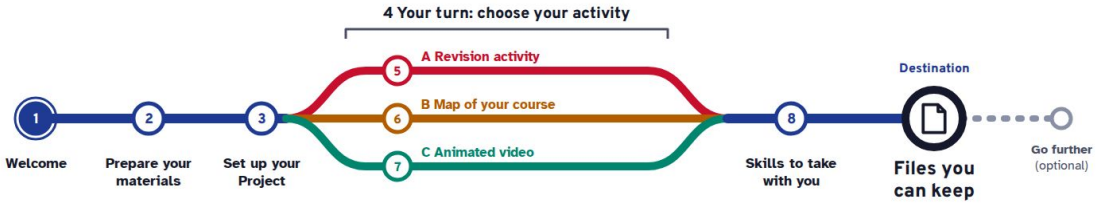
Go further (optional) and help

- Screen 9: How Claude reads your files, page 12
- Screen 10: Save, host and share, page 15
- Screen 11: Create a style guide, page 20
- Screen 12: Build, test, improve, page 21
- Screen 13: Examples for inspiration, page 23
- Screen 14: Troubleshooting, page 24

Screen 1: Use GenAI to make learning resources you can keep

Workshop worksheet

In this session you'll set up a **Claude Project with your own course materials**, then use it to make **learning resources for your students**, such as interactive web pages and animated explainer videos. You **don't need any coding experience**: you describe what you want, and Claude writes the code. Each resource is a **file you download and keep**, so you can continue using it whether or not the School keeps its Claude licence.



Your route through the session, ending with files you can keep. When it's your turn, choose one of three activity lines, A to C.

Things you need before you start

- An LSE Claude account. If you don't have one yet, request one with this form.
- If you have an LSE Claude account but see "Access denied" when you try to open shared resources, see Troubleshooting (screen 14).
- Three to five teaching files from one course, such as your course outline, one week's slides and a case study.
- Claude open in a second tab or window, so you can switch between it and this worksheet.

Your workshop leads

- **Dr Dori Sallai**, Associate Professor (Education) of Management, Department of Management
- **Steven Williams**, AI and Education Specialist, Eden Centre and Digital Skills Lab

Plan for the session

Time	What you'll do
10 min	Choose your materials and convert them to Markdown
10 min	Set up a Claude Project for your course
30 min	Build one resource: an interactive activity or an animated video
10 min	Plan where you'll use it, and

Time	What you'll do
	questions

Use **Next** at the bottom of the screen to move through the worksheet, or jump to any screen from the contents. Optional **Go further** screens at the end, on how Claude reads your files, saving, hosting and sharing, creating a style guide, building, testing and improving, and examples for inspiration, are for when you have more time.

Why does this session focus on creating portable resources?

There is currently uncertainty about the continuation of the Claude licence. Additionally, in general it's a good principle to build resources in a way that avoids platform lock-in. For these reasons, this session focuses on resources that don't depend on Claude. You will be shown how to create portable file formats, specifically HTML files for interactive activities, and MP4, GIF and PDF files for visual explainers. You can put these resources into Moodle or a free hosting site, and students will be able to use them with or without a Claude licence.

The skills you practise, such as giving GenAI tools the right context, asking for a plan before a tool creates a resource, and testing the final output before sharing, also work the same way in any other GenAI tool, whichever tools LSE ends up using.

What kind of resources does this session help you make, and where can you use them?

The file types you'll learn to make, and where each one can go. Students can open all of them without a Claude licence.

File type	Used for	Where it can go
HTML	Interactive activities	Moodle, Netlify, whatimade.app, GitHub Pages, Any web browser
MP4	Animated videos	OneDrive, YouTube or another video platform, PowerPoint slides, Moodle
GIF	Short looping animations	PowerPoint slides, Moodle, Web pages
PDF	Handouts and accessible versions	Moodle, OneDrive, Print

Screen 2: Prepare your materials

About 10 minutes

Preparing your materials means collecting the files you'll give Claude in one folder on your computer, then getting each file into the format that works best.

1. Collect your files in one folder

1. Decide which files to add: three to five from one course. Useful ones are:

- **your course outline,** with the topic for each week, so Claude can match a resource to where students are in the course
- **the slides or notes for one week** you'd like to build a resource for
- **a case study or seminar task** you already use
- **notes on what students find difficult,** such as common misconceptions or past exam feedback

2. Create a folder for this Project on your computer, for example in OneDrive, name it after your course, and put copies of the files in it.

This folder is your master copy. Later it will also hold your Project instructions, saved as a text file, your style guide if you make one, and every resource you download. Keep it up to date – whenever you edit or add new files to your Project, add them to this folder first, and then Claude. That way you won't end up with materials that only exist in Claude, which can be hard to get back out, and you'll have everything ready to use with another GenAI tool if you need to.

Check before you choose

- Leave out anything about individual students, such as names, marks or emails.
- Your own teaching materials are fine. For published readings and other third-party content, check the Library's Copyright Hub guidance on GenAI.

2. Prepare each file for its type

How you prepare a file depends on what's in it.

Files where an image or diagram is essential for understanding

Keep these as PDFs, so Claude can see the images. Claude Projects don't accept PowerPoint files, so save any slides like this as a PDF from PowerPoint first.

Everything else

Convert these to Markdown, a plain-text format that GenAI tools can read in full, using much less space than PDFs. You'll get better results, and you can open the files to see exactly what Claude sees. To find out more, see How Claude reads your files (screen 9) in the Go further section after the session.

1. Open deskmark (LSE Claude account required). It converts Word, PowerPoint and PDF files to Markdown inside your browser, so your files aren't uploaded anywhere.
2. Drag and drop your files into the deskmark upload box, then download the Markdown versions.
3. Save the Markdown files in your Project folder.

When your folder is ready, the next screen shows you how to set up a Claude Project and upload your files.

Screen 3: Set up your Project

About 10 minutes

1. In Claude, select **Projects** in the sidebar, then **New project**. Name it after your course, for example “MG317 Change Management resources”.
2. Select the pencil next to **Instructions**, paste in the template below and fill in the parts highlighted in [square brackets]. You may find it easier to paste the template into a Word document to edit it first, rather than editing it in the Instructions field.
3. Select **+** next to **Files** and add the Markdown files and PDFs from your Project folder. On some versions of Claude this section is called **Context**.
4. Check the capacity bar in that section. If it says Search mode, you may want to consider removing any PDFs and replacing them with Markdown versions.
5. Start a chat inside the Project and send the test prompt below.

Prompt: Project instructions template

```
## About this Project
I teach [course code and name], a [undergraduate year / MSc] course in
the Department of Management at LSE, with about [number] students.
I’ll use this Project to make learning resources for my students.

## My course
The files in this Project are my course materials. [File name] gives
the topic for each week. When you make a resource for a particular
week, only use ideas students will have met by that week, and link new
ideas to ones they’ve studied earlier.
Students often find these things difficult: [list them, or delete this
line].

## How to work with me
- Base all content on my materials. Flag any examples, facts or claims
you add that aren’t in them, so I can check them.
- Before building anything, ask me any questions you need, one at a
time. Then show me a short plan with the content, and wait for my
approval.
- Use British English, and plain, conversational language that speaks
to the student as “you”.

## Rules for every interactive resource
- Once style-guide.md is in this Project, follow it for every
resource.
- Make it a standalone Artifact: a single self-contained HTML file,
with all the styling and code inside it, so I can download it and host
it outside Claude, for example on Moodle.
- Don’t use Artifact storage, AI-powered features or anything else
that needs Claude to run. Don’t collect or send any information from
users. If I ask for answers to be kept between visits, store them only
in the student’s own browser.
- Don’t load images or files from other websites. Use Google Fonts
only, with a fallback font. Avoid external code libraries unless one
is really needed, and tell me if you use one.
- Make it understandable on its own, with a short introduction saying
what it is and how to use it.
```

```
- Make it work well on a phone and meet WCAG 2.2 AA: enough colour
contrast, full keyboard use, visible focus, labelled controls, no
meaning carried by colour alone, and respect for reduced-motion
settings.
- Remove any code comments that refer to our conversation.
- Before you finish, load it in a browser on your computer, click
through every screen and control, fix anything that doesn't work, and
tell me what I still need to check myself.
```

Prompt: Test prompt

```
What do students cover in week [number], and which ideas from earlier
weeks does it build on? Answer only from the files in this Project,
and tell me if anything is missing or unclear.
```

Why the instructions matter

Instructions apply to every chat in the Project, so you write your course context and rules once. They also build in two learning principles: introducing ideas in the order of the course, so students aren't overloaded (cognitive load theory, Sweller), and linking new ideas to earlier ones, so students build a connected picture of the subject (meaningful learning, Ausubel).

For more detail on Projects and Artifacts, see Creating effective Claude Artifacts – a guide for LSE educators (LSE Claude account required).

Screen 4: Your turn: choose your activity

You've set up your Project. Now choose one resource to build: an interactive activity (A or B) or an animated explainer video (C).

Each one takes about 30 minutes. Pick one that suits a course you teach; you can try the others later. For ideas, see Examples for inspiration (screen 13).

- **A. A revision activity**: Students practise concepts from several weeks, mixed together. See screen 5.
- **B. A map of your course**: Students explore how concepts connect across weeks. See screen 6.
- **C. An animated explainer video**: Students watch a concept build up step by step. See screen 7.

Have your own idea? You don't have to follow these guided activities. Start a new chat in your Project and tell Claude what you'd like it to build for your students. Ask it to check its plan with you before it starts. Your Project instructions still apply, so the result will be a file you can download and keep.

Screen 5: Activity A: A revision activity

About 30 minutes

Students practise recalling and applying concepts from several weeks of your course, mixed together, and explain their thinking before they see the answer.

This works for any course with concepts, frameworks or methods students need to remember and use, for example before an exam or at the end of a block of weeks.

Prompt: Build the activity

```
Using my materials for weeks [numbers], make a mixed revision activity
for students to use before [the exam / the end of term].

- Write 10 short questions on the concepts from those weeks. Each one
describes a short scenario, example or problem, and students decide
which concept explains it, or apply a concept to answer it.
- Mix the concepts up rather than grouping them by week.
- Start with a short explanation of why mixing topics feels harder but
helps students remember them for longer.
- Start with two of the clearest questions, then make them gradually
harder.
- For each question, students first type their answer and why, then
say how confident they are: Guessing, Unsure, Fairly sure or Certain.
- Then reveal a model answer with a short explanation of the clue that
points to it. If the student was confident but wrong, point this out
kindly.
- Finish with a summary of the concepts covered, showing which ones
the student found hardest, so they know what to revisit.

Show me the questions and explanations to check before you build it.
```

Prompt: Variation: a quick quiz on one week

```
Using my materials for week [number], make a short revision quiz for
students to use after the lecture.

- Write 6 questions that ask students to recall or explain the key
ideas in their own words, rather than choose from options.
- After each answer, show a model answer and the key points from my
materials, and ask the student to check their answer against them.
- End by asking students to write a two-sentence summary of the week,
then show a model summary to compare.

Show me the questions and model answers to check before you build it.
```

Learning principles in this activity

Mixing concepts from different weeks makes students work out which idea applies, which builds longer-lasting learning than practising one topic at a time (interleaving, one of Robert and Elizabeth Bjork's desirable difficulties). Typing an answer before seeing feedback is retrieval practice (Karpicke and Roediger), and rating confidence shows the gap between feeling familiar with an idea and understanding it. Starting with easier questions gives students early success, which builds their motivation to continue (Murayama).

Once it's built

Work through it as a student would. Then try a follow-up such as "Make question 4 harder" or "Add two questions on [a concept students often confuse]".

You can also add a feedback feature, so students can get detailed GenAI feedback on their answers. The activity gathers their answers with a ready-written prompt, which they paste into any GenAI chat. See **Let students get feedback from GenAI** on the Build, test, improve (screen 12) screen for the prompt to add it.

Screen 6: Activity B: A map of your course

About 30 minutes

Students explore the key concepts on your course as a network, seeing how each one builds on, extends or challenges the others. A test-yourself mode asks them to explain each concept before they read about it.

This is a smaller version of the idea behind LSE100's Disciplinary Atlas, and works well at the start of a course, before revision, or as a page students return to each week.

Prompt: Build the activity

```
Using the course outline and materials in this Project, make an
interactive map of the key concepts on my course.

- Show [8–12] concepts as points in a network, marked by the week
they're taught, with lines between concepts that are related. Label
each line with the relationship, such as "builds on", "extends" or
"challenges".
- When a student selects a concept, show a panel with a plain-language
definition, the week it's taught, the earlier concepts it builds on
and one concrete example from my materials.
- Add a "Test yourself" mode. Before a panel opens, the student types
an explanation of the concept as they'd give it to a friend with no
background in the subject. They then compare it with the definition
and a checklist of 3 things a good explanation includes.
- Include a list view of the same content, which students can switch
to, and which works well with a keyboard and screen reader.

Show me the list of concepts, definitions and connections before you
build it.
```

Prompt: Variation: recall cards

```
Using my materials for weeks [numbers], make a set of recall cards for
revision.

- Each card asks a question that needs a short written answer, such as
"Why might a change programme stall after early success?", rather than
a definition to recognise.
- Students type their answer, rate their confidence, then flip the
```

```
card to see a model answer and 2–3 key points to check their answer
against.
- Let students shuffle cards from different weeks together, and
explain briefly why mixing topics helps long-term memory.
- At the end, list the cards a student marked as Guessing or Unsure,
so they know what to revisit.

Show me the questions and model answers before you build it.
```

Learning principles in this activity

Seeing how concepts connect helps students organise what they learn into a coherent structure (schema development). Explaining a concept as you would to a beginner exposes gaps that rereading hides, and selecting and organising ideas in your own words turns information into understanding (generative learning, Fiorella and Mayer).

Without AI features, the page can't mark what students type. Instead, students compare their answer with a model answer and a checklist. The effort of recalling the idea stays with the student, and the page works anywhere. To let students get GenAI feedback as well, add a feedback package with the prompt on the Build, test, improve (screen 12) screen.

Screen 7: Activity C: An animated explainer video

About 30 minutes

Students watch a concept from your course build up step by step, in a short video they can watch before or after a lecture.

Claude can make short animated videos as MP4 files. It has no video-generation model: it draws each scene with code, captures the frames in a browser on its own cloud computer and joins them into a video. You can put an MP4 anywhere a video goes, including Moodle, a slide or your usual video platform.

This suits a concept that can be explained in a clear sequence, such as a process, a model or a framework.

For animated explainers made with Claude, including Dr Chris Blunt's clips for LSE100, see Examples for inspiration (screen 13).

What works well

Code draws flat shapes, diagrams, simple pictograms and text well, which suits processes, models and frameworks. It draws realistic people and detailed illustration poorly. Chris also notes that the technique suits conceptual visuals rather than charts of real data. For a picture code can't draw, you can generate one in Microsoft Copilot (sign in with your LSE account) and give it to Claude.

In the session

Good videos come from reviewing a storyboard and still frames before Claude renders the full video, and rendering takes several minutes. Aim to get as far as the storyboard and preview

stills today. The full render can carry on while you move to the next screen, as long as you leave the Claude tab open.

Prompt: Make an explainer video

```
I'd like a short animated MP4 video, about [60–90] seconds long,
introducing [concept] for students to watch before the week [number]
lecture. Base it on my slides for that week in this Project, keeping
my key terms and wording where they fit.

Before building anything:
- Interview me one question at a time: what students should be able to
do afterwards, what they already know, and what they find hardest.
- Suggest two or three visual styles that suit this subject and
recommend one. Code draws flat shapes, diagrams, pictograms and text
well, but not realistic people or detailed illustration, so don't
suggest styles that depend on those.
- Give me a scene-by-scene storyboard with the on-screen text for each
scene. Calculate the timings in code: at least one second for every
three words plus two seconds, then hold each finished scene for three
more seconds. If the content won't fit the length, tell me what to
cut. Flag anything you've added that isn't in my slides.

When building:
- There's no narration, so all the words are on screen. Put labels
next to the parts they describe, and highlight the part the text is
talking about.
- Build any diagram up one part at a time, in the order the text
introduces them, rather than showing it all at once.
- Bring in each piece of text with the visual it describes, and don't
start any movement until the text has been read.
- No music or sound effects.
- Show me still frames of each scene before you render the full video,
so I can make changes first.
- Make it a 1080p MP4. Once I've approved it, make a text version for
students who can't watch it.
```

Prompt: Variation: an animated GIF for a slide

```
Make a short animated GIF showing [one idea, for example how
resistance to change builds and falls] for a slide in my week [number]
lecture. I've uploaded a screenshot of one of my slides so you can
match its colours and fonts.

- Keep it to one idea, with labels large enough to read when
projected.
- Keep it under five seconds, or have it play once and stop on the
final frame, and hold the final state for a couple of seconds.
- Give me alt text describing what changes in the animation, and a
still of the final frame for handouts.
```

Prompt: Variation: visuals for a lecture video you've already recorded

```
I've uploaded the captions file (VTT or SRT) for a lecture video I've
recorded. Make a silent clip of visuals for the part where I explain
[topic], to replace the footage for that section.

- Find the passage in the transcript and confirm its start and end
times with me.
- Bring each visual in as my narration first mentions it, a moment
early rather than late.
- Because I'm speaking the words, keep on-screen text to key words and
labels.
- Make the clip exactly the length of the passage, at 1920 × 1080 and
25 frames per second.
- Give me a timing sheet listing each visual, the words that cue it
and their time.
```

Learning principles in these prompts

The prompts follow Richard Mayer's principles of multimedia learning: words next to the pictures they describe, highlighting where to look, building diagrams up in stages, and leaving out anything decorative, including background music, which can reduce learning. Because these videos have no narration, the timing rules give students enough time to read each piece of text before anything new appears.

For more detail, see Creating effective Claude Artifacts – a guide for LSE educators (LSE Claude account required).

Screen 8: Skills to take with you

About 2 minutes

Everything you practised today works in any GenAI tool:

- **Give it context.** Start from your own materials and course sequence.
- **Use plain-text files.** Markdown takes little space, and you can see exactly what the tool sees.
- **Ask for a plan before a build.** Have the tool ask you questions and show you the content first.
- **Check what it makes.** Compare the content with your materials and test it as a student would.
- **Keep the files.** Download what you make and host it yourself.

Where will you use your resource first?

Further resources

- Creating effective Claude Artifacts – a guide for LSE educators (LSE Claude account required)
- Guidance on using Claude – Eden Centre videos and guides on Artifacts and Projects

- AI & Education Toolkit – step-by-step guides based on LSE educators' practice (LSE login)
- Using Claude Code and Artifacts to empower student learning – Dr Casey Kearney (School of Public Policy) on students making Artifacts to explain course concepts and sharing them on Moodle
- Use Artifacts with a keyboard or screen reader – Anthropic's guide, which you could share with students

Go further when you have time

Five optional screens follow this one. They aren't needed in the session, but they'll help you share your resources, make them better and find new ideas.

- **How Claude reads your files**: Why the worksheet asks you to prepare files in Markdown. See screen 9.
- **Save, host and share**: Put your HTML file on Moodle or a free hosting site. See screen 10.
- **Create a style guide**: Keep everything you make looking and sounding consistent. See screen 11.
- **Build, test, improve**: Follow-up prompts, testing tips and GenAI feedback for students. See screen 12.
- **Examples for inspiration**: Interactive activities and visual explainers made by LSE colleagues. See screen 13.

Questions about today

- **Dr Dori Sallai**, Workshop lead, Department of Management, d.sallai@lse.ac.uk
- **Steven Williams**, AI and Education Specialist, Eden Centre and Digital Skills Lab, s.williams10@lse.ac.uk

Help using GenAI in your teaching

- **Jamie Hayward**, AI and Education Specialist, Eden Centre and Digital Skills Lab, for support with using GenAI in your courses, J.Hayward1@lse.ac.uk

Screen 9: How Claude reads your files

Go further (optional)

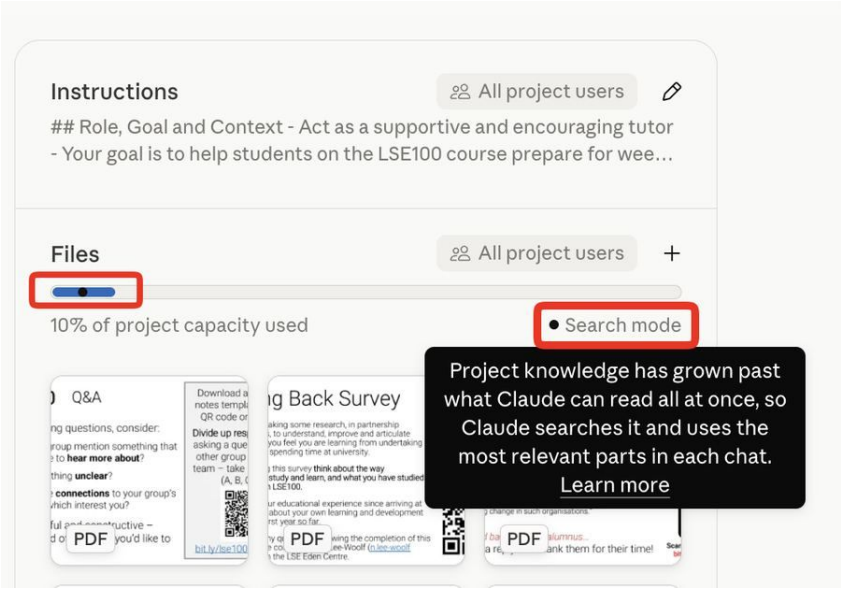
This screen explains why the worksheet asks you to prepare your files in Markdown. You don't need to read it during the session, but it's useful background for later.

A Claude Project is a workspace that holds files and instructions. Claude uses them in every chat you start inside that Project. How you add your files affects how well Claude can use them.

Two ways Claude reads a Project

Everything in memory. While the files are small enough, Claude reads all of them at the start of every chat. They sit in its context window – the working memory of a large language model – so Claude can draw connections across your whole course.

Search mode. Once the files grow too large, the Project switches to Search mode. Claude then looks up the snippets that seem most relevant to each question, so it can miss context or nuance elsewhere in your materials.



A Project that has switched to Search mode at only 10% of its capacity. All the files shown are PDFs.

PDFs are the usual cause. Claude keeps each PDF page as an image as well as text, so a PDF takes up far more space than the same content as plain text.

What is Markdown?

Markdown is a way of showing formatting in plain text. A Word document stores its formatting in hidden code. Markdown uses a few punctuation marks you can see instead: a # at the start of a line makes it a heading, and two asterisks either side of words make them bold. Markdown files end in `.md`, and you can open them in any text editor, such as Notepad or TextEdit.

What the Markdown file contains:

```
# Week 3: Leading change

**Kotter's eight steps** begin with:

- Create a sense of urgency
- Build a guiding coalition
```

What it means:

A heading, "Week 3: Leading change", followed by the sentence "Kotter's eight steps begin with:", with "Kotter's eight steps" in bold, and a bulleted list of two items: "Create a sense of urgency" and "Build a guiding coalition".

Because Markdown is plain text, it takes up very little space, and you can open the file to see exactly what Claude sees. It has become the standard format for GenAI tools, so the same files work in Copilot or any other chatbot. You don't need to write Markdown yourself: the deskmark converter on the Prepare your materials (screen 2) screen does it for you.

Which format to use

What you have	Add it as
Word documents with no important images	Markdown. You can add Word files directly, but Claude converts them to Markdown and drops any images.
PowerPoint slides	Markdown. Projects don't accept PowerPoint files.
Slides or readings where a diagram or image matters	PDF, but only for those files.
Spreadsheets	CSV, which is also plain text.

Good advice for any GenAI tool

Search mode is specific to Claude Projects, but the same advice works for Copilot, ChatGPT and other GenAI chatbots. Most of them convert the files you upload into text behind the scenes, and large files such as PDFs use up the space they can read at once. Giving any tool your materials in Markdown, and keeping PDFs for files where images matter, means it reads your content in full and you can check exactly what it's working from.

Quick check

Type your answer and choose how confident you are before you reveal ours. These are just exercises to help you check your understanding, so nothing you type is saved.

1. You want to add ten weeks of lecture slides in PowerPoint. Only week 4 contains a diagram students need to understand. What would you add to the Project?

Suggested answer:

Convert all ten weeks to Markdown, and add week 4 as a PDF as well, so Claude can see the diagram. Projects won't accept the PowerPoint files, and ten PDFs would use far more capacity than one.

Compare it with your answer. What did you include, and what did you miss? If you chose Certain and missed something, that's worth noticing.

2. Your Project has switched to Search mode. What could you change?

Suggested answer:

Replace any PDFs that don't contain important images with Markdown versions, and remove files Claude doesn't need for the resources you want to make. Watch the capacity bar: once the Search mode label disappears, Claude is reading everything again.

Compare it with your answer. What did you include, and what did you miss?

Why the quick check works this way

Typing an answer makes you recall the idea rather than recognise it, which strengthens learning more than multiple choice (the retrieval-practice effect, Karpicke and Roediger). Rating your confidence first shows whether you knew the answer or just found it familiar. You can build both features into resources for your students.

Screen 10: Save, host and share

Go further (optional)

This screen is about sharing HTML files, such as the interactive activities, which takes a few more steps than other files.

Sharing videos, GIFs and PDFs

Share these as you would any other file. Put an MP4 video on OneDrive, YouTube or another video hosting service and share the link, or add it straight to a slide. A GIF can go straight into a slide, and a PDF can be uploaded to Moodle.

Download your file

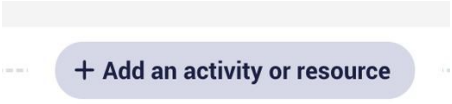
Select the Artifact's name at the top left and choose **Export** to download it as an HTML file. Keep the master copy somewhere safe, such as OneDrive. To change it later, upload the file to a new chat – in Claude or another GenAI tool – and describe what you want changed.

Before sharing, open the downloaded file in your browser to check it works on its own. The guide Creating effective Claude Artifacts – a guide for LSE educators has more on exporting and sharing.

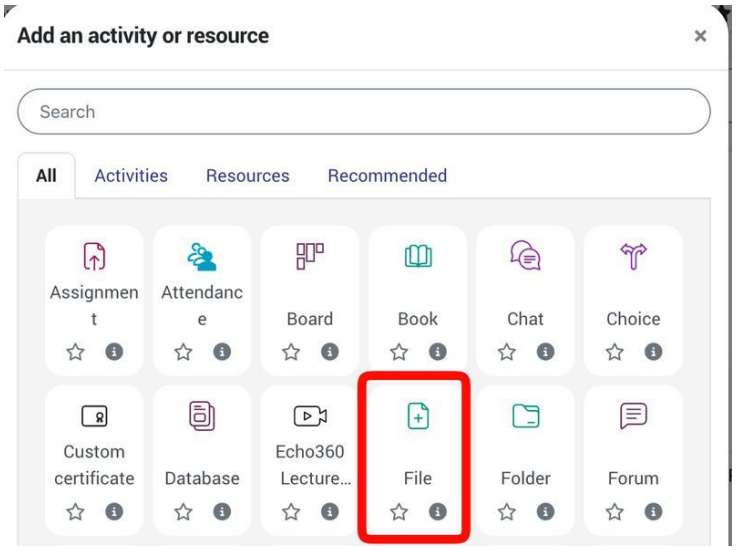
Option 1: Moodle

Students open it with their usual LSE login, and the content stays behind that login.

1. On your Moodle course page, switch on **Edit mode** at the top right.
2. In the section where the resource should go, select **Add an activity or resource**.

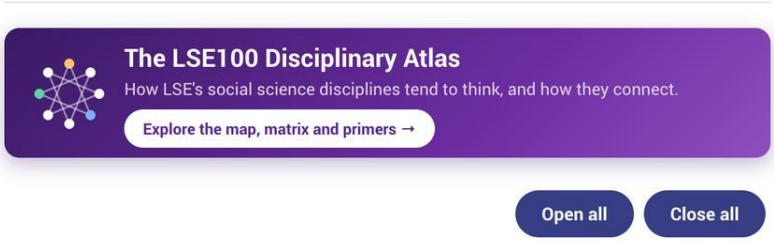


3. Choose **File**.



4. Give it a name, then drag your HTML file into the upload box, or select it from your computer.
5. Under **Common module settings**, choose an **Availability** option:
   - **Show on course page** – students see it in the section.
   - **Hide on course page** – students can't open it.
   - **Make available but don't show on course page** – students can open it only from a link you give them.
6. Select **Save and return to course**.

Dr Chris Blunt uses the third option, then links to the file from a **Text and media area**, which lets him make a more visual link. To get the file's link, open it from your course page while Edit mode is on and copy the address from your browser.



Chris's link to the Disciplinary Atlas, made in a Text and media area. You can also upload the file and leave it on the course page.

Add an accessible PDF version

Some students who use assistive technology, such as a screen reader, may find an interactive page hard to use. Upload a PDF version of the same content alongside the HTML file, and link to both. You can ask Claude to make the PDF with the prompt below, then check it before you share it, using Adobe Acrobat's accessibility checker or PAC (PDF Accessibility Checker), a free tool that tests whether a PDF meets the PDF/UA and WCAG accessibility standards. PAC only

runs on Windows. On a Mac, use axesCheck, a free online checker from the same company, which tests a PDF you upload.

Prompt: Make an accessible PDF version

```
Make an accessible PDF version of this resource for students who use
assistive technology, such as a screen reader, and may find the
interactive version hard to use. Make it a separate file I can upload
next to the HTML file.

Content
- Include all the content: the introduction, every explanation,
question and answer, and a written description of each visual and of
what changes when students use the controls.
- Rewrite any interactive tasks so they work in a document, for
example by listing the options and putting the answers in a separate
section at the end, so students can try each one before checking.
- Turn any diagram that shows which item goes with which into a table
with header cells.
- Leave out buttons, text boxes and other controls, and don’t add
anything that isn’t in the resource.

Structure
- Use one Heading 1 for the title, then headings in order without
skipping levels. Keep any labels, such as timings, after the heading
they belong to.
- Give every image alt text, use real lists, and give tables header
cells.
- Leave out decorative images, such as photos of people, and give
every other image a set width, no wider than the text and smaller for
small screenshots, so images don’t print at their full pixel size.
- If the document is longer than a few pages, add a contents list near
the start, with each entry linking to its heading and giving its page
number.

How to make the file
- Don’t print the web page to PDF from a browser. Browser PDFs leave
backgrounds and borders untagged and usually fail accessibility
checkers.
- Instead, write the content as a simple, semantic HTML file and
convert it to Word with pandoc, setting the language to British
English (en-GB). Set the document title in the Word file’s properties,
not as an extra title paragraph above the Heading 1.
- Rename any Word paragraph style called “Figure”, because LibreOffice
turns style names into PDF tags and a style called Figure clashes with
the real image tags.
- Export the Word file to PDF with LibreOffice, with PDF/UA
compliance, tagged PDF and bookmarks turned on. If LibreOffice or
pandoc isn’t available, tell me before using another method.

Checks
```

```
- Check in code that every piece of page content is either tagged or
marked as decoration, the language is en-GB, the title is set and
shown in place of the file name, and the PDF/UA identifier is in the
metadata.
- Also check that headings are in order, every image has alt text,
tables have header cells, every link has a description and goes to the
right place, the fonts are embedded, and no standard tag names have
been remapped to other tags.
- Set every text colour, including headings and links, to a dark
colour that meets WCAG 2.2 AA contrast. Pandoc's default Word styles
use a light blue for headings and links that fails, so replace it in
the styles, and check the contrast of every text colour in the
finished PDF in code.
- Tell me anything you couldn't check, so I can test the PDF with
axesCheck or PAC. If it fails, I'll give you the report so you can fix
it.
```

For examples of PDF handouts made this way, see Examples for inspiration (screen 13).

Option 2: Netlify Drop

Gives you a public web address that anyone can open, with no login.

1. Rename your file `index.html` and put it in a folder of its own.
2. Go to Netlify Drop and drag the folder onto the page.
3. Sign up for a free account when prompted. Without one, the site is deleted after an hour.
4. In the project's settings, change its name to set a readable address, such as `mg317-change-quiz.netlify.app`.
5. To update it later, drag the updated folder onto the deploy area of your project.

Example: Dr Dori Sallai's AI Policy Mentor.

Option 3: whatimade.app

A free hosting service made by an LSE student. It also gives you a public web address.

1. Go to whatimade.app and drag your HTML file onto the page.
2. Sign in with a Google account when prompted. A copy of each version is saved in your Google Drive.
3. Your resource goes live at an address ending in `.whatimade.app`.

Example: Lourdes Hernández-Martín's GenAI Starter Guide.

Option 4: GitHub Pages

GitHub is a free service for storing and sharing files, and GitHub Pages turns a folder of files into a public website. It's a bit more technical than the other options, with more settings to find, but it's free, reliable and widely used. You can do everything in your web browser by uploading files, with no need for the command line.

1. Rename your file `index.html`.

2. Sign up for a free account at GitHub, or sign in.
3. Select **+** at the top right, then **New repository**. A repository is GitHub's name for a project folder. Give it a short name, such as `change-quiz`, set it to **Public**, and select **Create repository**.
4. On the next page, select **uploading an existing file**. Drag in `index.html`, along with any other files it uses, then select **Commit changes** to save them.
5. Select **Settings** at the top of the repository, then **Pages** in the left sidebar under "Code and automation".
6. Under **Build and deployment**, set **Source** to **Deploy from a branch**. Choose the **main** branch and the **/ (root)** folder, then select **Save**.
7. After a minute or two, refresh the page. Your web address appears at the top, in the form `yourusername.github.io/change-quiz`.
8. To update it later, open the repository, select **Add file**, then **Upload files**, and upload the new `index.html`. The site updates within a few minutes.

On a free account, the repository has to be public for GitHub Pages to work. Anyone can see the files in it as well as the web page, so the advice on public links below applies here too. Your GitHub username also appears in the web address.

Option 5: Lovable, for resources that need GenAI inside them

The resources you make in this session don't use GenAI once they're downloaded, because a file on its own has no connection to an AI model. If your activity needs GenAI while students use it, for example a dynamic role-play where an AI character responds to what students write, you can build and host it on Lovable instead. Dori's and Haider's examples in the first part of the session are hosted there.

Lovable apps can use AI models from Google, OpenAI and Anthropic. Lovable picks one by default, or you can name the model you want when you describe the feature. You don't need your own API key, because the AI use comes out of your Lovable credits.

- **Free account:** 5 credits a day for building, up to 30 a month, plus a small monthly allowance for AI features inside your apps. That's enough to try an AI feature, but not to run one for a whole class. Your app gets a `.lovable.app` web address and a small Lovable badge.
- **Paid accounts:** from $25 a month, with 100 monthly credits, AI use inside your apps charged as you go, your own domain and no badge. Costs for AI features grow with the number of students using them, so check your usage in Lovable's settings.

Plans change often, so check Lovable's pricing page for the current limits.

1. Sign up at lovable.dev.
2. Start a new project and describe what you want to build. You can also paste in the code of a resource you made in Claude and ask Lovable to rebuild it.
3. Describe the AI feature you need, for example "Add a role-play where students interview the CEO from my case study, and the AI answers in character".
4. Test it as a student would, then select **Publish** to get a web address you can share.

Lovable isn't supported by LSE, and anything students type into an AI feature is sent to the AI model. Don't ask students to enter personal information, and check the guidance on public links below.

Public links

You need an account to upload to Netlify, whatimade.app, GitHub or Lovable, but students only open a link. They don't sign up or give any data, which fits the Eden Centre's guidance on using unsupported third-party tools.

Anyone with a public link can see the page, so only publish your own materials, with nothing about students and no third-party content you don't have the rights to share. Use Moodle for anything that should stay behind the LSE login.

Wherever you host it, put the accessible PDF version next to the link, using the prompt above.

Screen 11: Create a style guide

Go further (optional)

A short style guide keeps the resources you make for a course looking and sounding consistent, so you don't have to describe your preferences in every prompt.

1. Start a new chat in your Project and send the prompt below. Claude asks you questions one at a time and uses your course materials to suggest styles.
2. Check the sample page Claude makes, and ask for any changes.
3. When you're happy with it, download `style-guide.md` and add it to your Project's **Files** (**Context** in some versions of Claude).
4. Start each new resource in a new chat in the Project. The instructions template already tells Claude to follow your style guide.

Prompt: Style guide interview

```
I'd like a short style guide for the learning resources I make for
this course, so they look and sound consistent. Interview me one
question at a time, drawing on my course materials in this Project,
about:
- the look and feel I want (if I don't have one in mind, suggest two
or three styles that draw on the visual world of my subject, and
recommend one)
- colours, checking in code that text colours meet WCAG 2.2 AA
contrast
- fonts, using Google Fonts with fallbacks so they work in downloaded
files
- tone of voice
- anything that should appear on every resource, such as the course
code or a credit line
- anything to avoid

Then show me a one-screen sample page in this style. Once I approve
```

```
it, write the style guide as a Markdown file called style-guide.md, no
longer than one page, for me to add to this Project.
```

Why a separate file?

Your style guide is a plain-text Markdown file, so you can open and edit it yourself, reuse it in other Projects and take it to any GenAI tool.

Logos, images and templates

- **Logos.** Check LSE's brand guidelines before using the LSE logo. On a public web page it can suggest the resource is an official School publication, and many teaching resources don't need one.
- **Images.** When you want an image in a resource, attach it in the chat where you're building and ask Claude to include it in the file, so the download works anywhere. Each image can be up to 20 MB, or 2 MB for an SVG file, though these limits may change. Use images you have the rights to, such as your own, free stock images with attribution, or images generated in Microsoft Copilot. Keep them in a folder on OneDrive and list their file names in your style guide.
- **Templates.** If you have a resource whose look you like, attach it during the interview and ask Claude to describe its style in the guide. You don't need to keep the original in your Project.

For more detail, see Creating effective Claude Artifacts – a guide for LSE educators (LSE Claude account required).

Screen 12: Build, test, improve

Go further (optional)

Claude rarely gets a resource right first time. These habits help, in Claude or any other GenAI tool:

- **Start from your materials.** Prompts that point to your own files give more accurate, course-specific results than asking Claude to write content from scratch.
- **Check the content before the design.** The prompts on the Activity A to C screens ask Claude to show you the content first, such as the scenarios and explanations, or a storyboard for a video, before it builds anything. If you're building something of your own, ask for this too.
- **Change one thing at a time.** Test the first version, then give one follow-up instruction per message.
- **Ask for a standalone Artifact.** For simple requests, Claude sometimes draws a visual inside the chat instead. If that happens, ask it to "change it to a standalone Artifact".
- **Test it as a student would,** on a phone as well as a laptop.

Useful follow-up prompts

- Check every scenario and explanation against my materials. List anything you added that isn't in them, so I can check it.
- The text overflows its box on a phone screen. Fix it.

- Make a Word version of all the content, with a clear heading structure, for students who find the interactive version hard to use.
- Load the Artifact in a browser on your computer, click through every screen and control, fix anything that doesn't work, and tell me what I need to check myself.

For more tips, see Creating effective Claude Artifacts – a guide for LSE educators (LSE Claude account required).

Let students get feedback from GenAI

A resource that runs without Claude can't mark what students write. One way round this is a **feedback package**: a button that copies the student's answers together with a ready-written prompt, which they paste into any GenAI chat for targeted feedback.

The prompt below adds a feedback package to any resource you've built. Send it as a follow-up once the rest of the resource works.

Prompt: Add a feedback package

```
Add a "Get feedback" step to the end of this resource, so students can
get feedback from any GenAI tool without the resource needing Claude
to run.

- Add a final screen with a "Copy feedback package" button. It copies
a ready-written prompt, followed by everything the student has written
and chosen, including any confidence ratings. If copying is blocked,
select the text so students can copy it themselves. Add a "Download as
a text file" button as a backup.
- Tell students to paste the package into a new chat in an LSE-
supported GenAI tool, such as Claude or Copilot signed in with their
LSE account, and not to add any personal information.
- Keep students' answers only in their own browser, so they can come
back to the resource later, and add a "Clear my answers" button.
- Write the prompt in the package so the GenAI tool acts as a tutor
and structures its feedback around three questions (Hattie and
Timperley):
 - Where am I going? Restate the learning goal and success criteria
for this activity, which you'll include in the package from my
materials.
 - How am I going? Compare the student's answers with the success
criteria, pointing to specific parts of what they wrote, and compare
their confidence ratings with how well they did.
 - Where to next? Give one or two specific next steps, most important
first.
- In the prompt, tell the tool not to rewrite the student's answers or
give model answers, to say when it's unsure what a reading argues
instead of guessing, to keep the feedback to about [300] words in
British English, and to end by asking one question about the student's
weakest area, then wait for their reply.
- Include a short summary of the relevant course content from my
materials in the package, so the feedback matches the course.
```

```
- After the copy button, add a box where students write one thing
they'll do next, and include it in the downloaded file.

Show me the full prompt, success criteria and course summary that will
go in the package before you build it.
```

Why three questions

Feedback works best when it answers three questions for the learner: Where am I going? How am I going? Where to next? (Hattie and Timperley, 2007). Building them into the prompt means any GenAI tool gives feedback against your success criteria, and ends with a next step the student can act on.

Screen 13: Examples for inspiration

Go further (optional)

Resources made with Claude by LSE colleagues. You probably won't have time to look through these during the session, so come back to this screen at any time for ideas.

Examples of interactive activities made with Claude

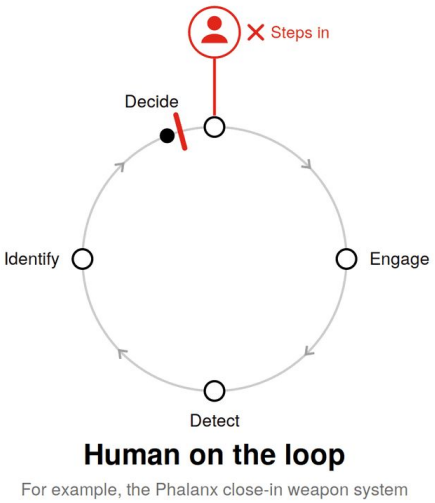
- AI in Education Tools – Dr Dori Sallai's tools for Management students and faculty, several hosted on Netlify and open to anyone with the link.
- GenAI Starter Guide – Spanish Language and Society – Lourdes Hernández-Martín's student guide (Centre for Language Studies), hosted on whatimade.app.
- The Disciplinary Atlas – made with Claude by Dr Chris Blunt for LSE100. It's a single HTML file uploaded to the LSE100 Moodle page; the link opens a copy so you can try the interactive map (LSE login). There's a screenshot of it on the Welcome screen.

Examples of visual explainers made with Claude

Steven Williams made these explainers with Claude. Each topic has an MP4 animated video, an HTML interactive explainer and an accessible PDF handout.

Topic	Video	Interactive	Handout
Jeremy Bentham's Panopticon	Watch the video	Explore the 3D model	Open the PDF
The tragedy of the commons	Watch the video	Try the interactive	Open the PDF
Exponential growth (rice and the chessboard)	Watch the video	Try the interactive	Open the PDF

- Building animated explainers for lecture videos with Claude – Dr Chris Blunt's blog post, with five example clips made for LSE100 videos.



Human on the loop, an animated GIF made with Claude by Dr Chris Blunt for LSE100. A GIF like this can go straight into a slide.

Screen 14: Troubleshooting

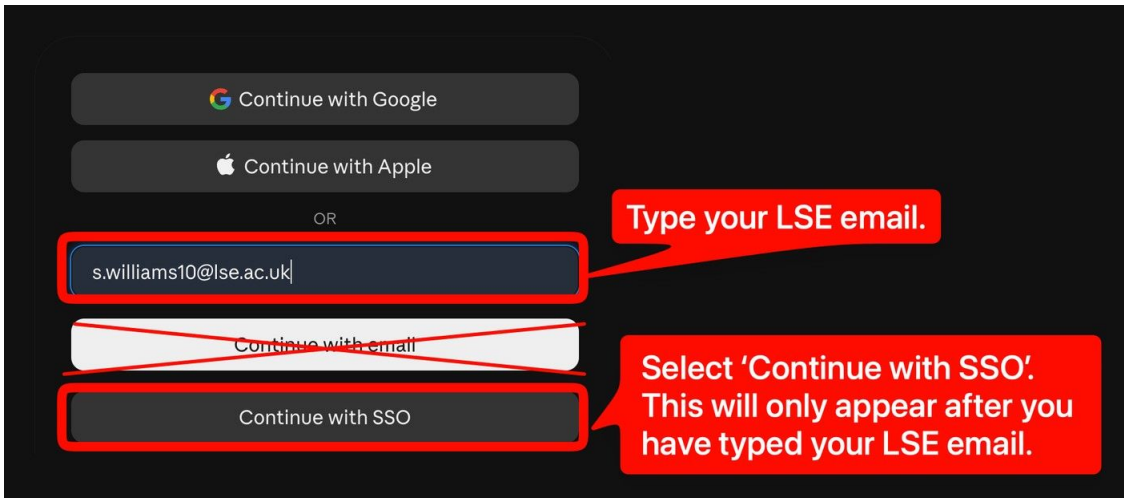
Help

"Access denied" when opening shared resources

If you've requested an LSE Claude account with the request form but still can't open resources other people have shared, the most common cause is that you aren't signed in with SSO (single sign-on). This means your account isn't linked to LSE's version of Claude.

How to sign in with SSO

1. Sign out of Claude.
2. Go to claude.ai and type your LSE email address in the email box.
3. Don't select **Continue with email**. A **Continue with SSO** button appears: select that instead.
4. You'll be taken to a Microsoft sign-in screen. Enter your LSE email address again and your normal LSE password.



Type your LSE email address, then select Continue with SSO rather than Continue with email.

If you still can't get in, email the IT service desk at tech.support@lse.ac.uk.