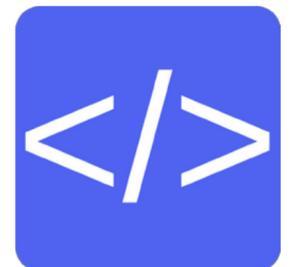
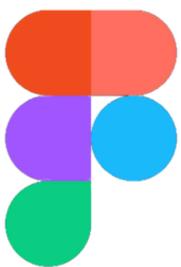


CHATATHON

VIRTUAL LIVING LAB



The Chatbot Hackathon Virtual Living LAB is a creative, hands-on learning event where small teams of students design and build their own simple chatbots.

A hackathon is like a short, intensive workshop where people come together to solve problems, build prototypes, and share ideas. Instead of competing against each other, this hackathon focuses on learning-by-doing, teamwork, and creativity.

In this hackathon:

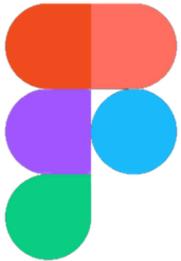
- Students work in teams of 4–6 people
- Each team chooses a real-world vocational challenge (e.g., bakery, clothing repair, bike shop, car service)
- Teams design a conversation flow for their chatbot using FigJam
- With the help of ChatGPT, they transform their flow into working chatbot code (HTML, CSS, JavaScript)
- They test and improve their chatbot using free online coding platforms (OneCompiler).

The hackathon can be run as a live event (one full day with 20–30 participants per partner) or as a “hackathon-in-a-box” – a ready-to-use kit that teachers can download and run anytime in their classroom.

At the end of the hackathon, every team will have:

1. A chatbot flowchart (in FigJam)
2. A working chatbot prototype (tested online)
3. A project description (submitted via Google Form)

CORE TECHNOLOGIES



FigJam (Figma's Online Whiteboard) found @ www.figma.com/figjam

FigJam is a collaborative digital whiteboard used for brainstorming, diagramming, and mapping ideas.

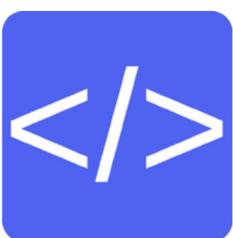
Before coding, teams must plan their chatbot conversations. FigJam allows them to draw a flowchart with sticky notes and arrows.



ChatGPT (AI Text Generator) found @www.chatgpt.com

ChatGPT is an AI tool that can understand text instructions and generate code, explanations, or creative writing.

Why we use it: Students don't need to be coding experts. Once they have their flowchart, they can copy it into ChatGPT and ask it to generate chatbot code (HTML, CSS, and JavaScript).



OneCompiler (Online Code Editors) found @www.onecompiler.com

Free websites where you can paste code and instantly see how it works. No installation is required.

Why we use them: Once ChatGPT generates the chatbot code, students paste it into these editors to test it. They can see their chatbot in action, type questions, and refine answers.

FIGMA JAM WHITEBOARD

FigJam (Figma's Online Whiteboard)

How students will use FigJam

Each team will receive a FigJam Starter Template with:

- **Yellow sticky notes** = User questions (e.g., "What cakes do you have?")
- **Blue sticky** notes = Bot answers (e.g., "We have chocolate, vanilla, and carrot cakes.")
- Arrows to connect user notes to bot notes

Student task:

1. Duplicate sticky notes for every question/answer they want to add
2. Connect them with arrows to show the path of the conversation
3. Build question-answer pairs to make a useful chatbot.

Example (Bakery Bot mini-flow in FigJam):

User: "When are you open?"

Bot: "We're open Mon-Sat, 8:00-18:00."

User: "Do you have gluten-free bread?"

Bot: "Yes! We bake gluten-free bread every morning."

User: "What cakes do you sell?"

Bot: "We have chocolate, vanilla, and carrot cakes."

Teacher tips for FigJam stage:

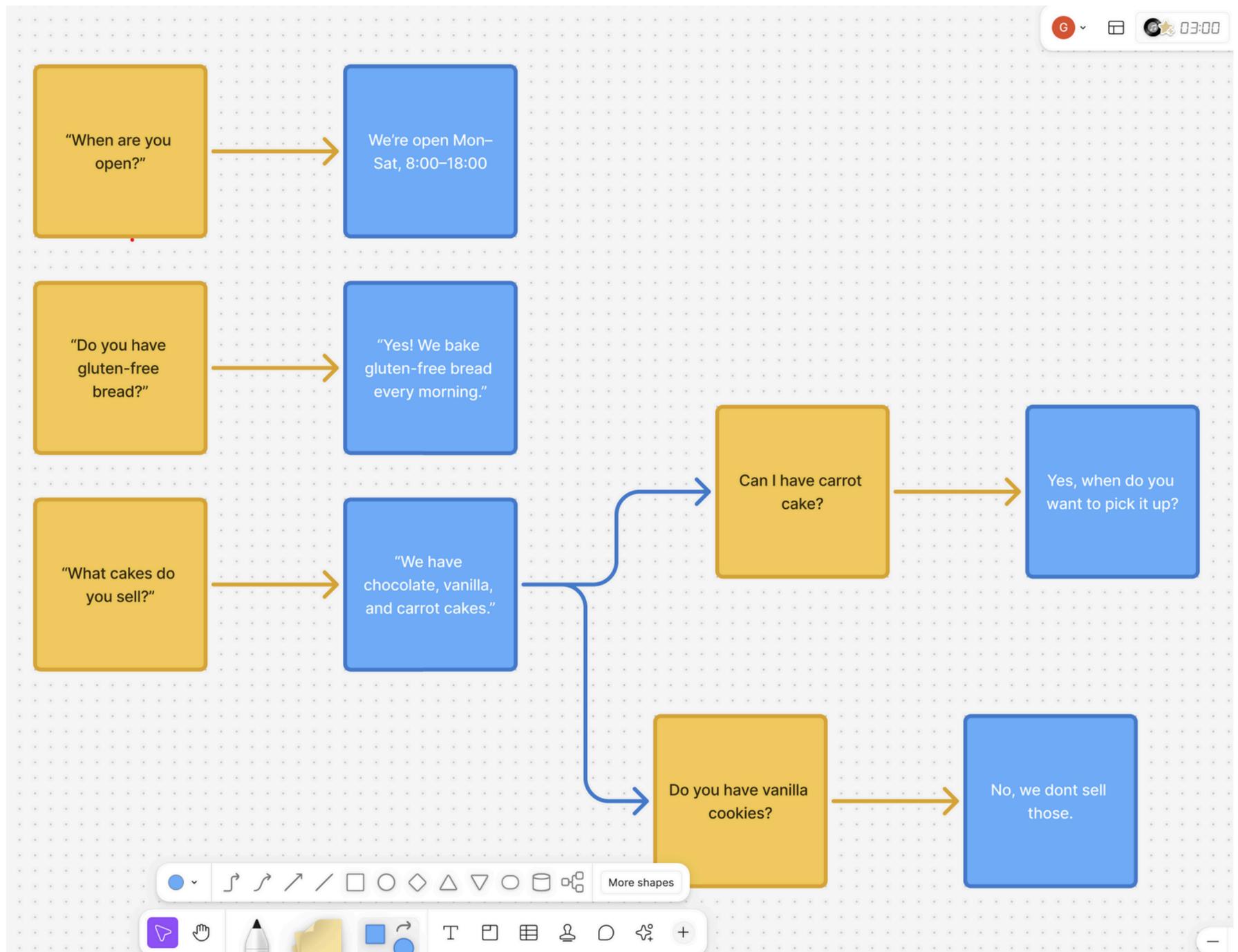
Encourage students to start simple: 3-4 questions and answers is enough to begin. They can expand later.

Remind them that chatbots don't need to know everything. It's better to focus on a small, realistic problem.

Walk around and check if arrows are connected clearly. Teams often forget to show direction in flows.

If teams get stuck, suggest a default fallback answer: "Sorry, I don't understand that yet."

FIGMAJAM WHITEBOARD



FigJam (Figma's Online Whiteboard)

The flowchart can be as big as you want and should cover the topic/idea as best as possible.

After the flowchart is done, you have to export it as a PDF and upload it to **ChatGPT** for writing the code.

Note: if you are doing the task as an online event, many students can work on the same flowchart at the same time.

CHATGPT DEBUGGING

ChatGPT (AI Text Generator)

Step 1 – Export from FigJam

- In FigJam, once your team is happy with the chatbot flow, go to:
- Menu → File → Export → PDF
- Save the flowchart as a PDF file on your computer.

Step 2 – Upload to ChatGPT

- Open ChatGPT → Upload your flowchart PDF.
- Use this first prompt:

“Here is a chatbot flow we created in FigJam. Please read the user questions and bot answers from this flowchart and rewrite them as a structured list of question–answer pairs. If something is unclear, ask me.”

Step 3 – Verify the Question–Answer List

- ChatGPT will return a text version of your chatbot’s logic.
- Carefully check:
 - Are all questions included?
 - Are all answers correct?
 - Did ChatGPT mix up any arrows or connections?

Step 4 – Debugging Misunderstandings

It’s common that ChatGPT may misinterpret parts of the PDF. Examples of errors:

- It skipped a question
- It linked the wrong answer to a question
- It merged two steps into one

Example fix prompt:

“You missed one path: if the user asks ‘Do you sell gluten-free bread?’, the bot should answer ‘Yes, every morning.’. Please add this to the list.”

or

“You connected ‘What cakes do you have?’ to the wrong answer. It should be ‘We have chocolate, vanilla, and carrot cakes.’. Please fix that.”

Step 5 – Confirm the Final Flow

- Once ChatGPT updates the Q&A list, review it again as a team.
- When the text version matches your FigJam flow perfectly, you are ready to move to coding.

CHATGPT CODING

ChatGPT (AI Text Generator) + OneCompiler (Online Code Editors)

Step 1 – Open OneCompiler and Select HTML

OneCompiler is a free online coding environment where students can write and run code directly in their browser. No installation is required, which makes it perfect for classroom hackathons.

For this activity, students must use HTML mode, because we want all code (HTML, CSS, JavaScript) inside one single file. This ensures ChatGPT's outputs will run smoothly without splitting into separate files.

Student Instructions

1. Open your browser and go to <https://onecompiler.com>.
2. From the list of languages and templates, choose HTML, CSS, JS.
3. The screen will open with a default project that has three files: index.html, style.css, and script.js.

Important Adjustment

- We don't want separate files for this challenge.
- Ask ChatGPT to always return a single HTML file with inline `<style>` and `<script>`.
- Students will delete everything in the editor and replace it with ChatGPT's code.



```
index.html  styles.css  script.js  +  [ ]  ✎
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Hello, World!</title>
5  <link rel="stylesheet" href="styles.css" />
6  </head>
7  <body>
8  <h1 class="title">Hello World! </h1>
9  <p id="currentTime"></p>
10 <script src="script.js"></script>
11 </body>
12 </html>
```

CHATGPT

CODING

ChatGPT (AI Text Generator) + OneCompiler (Online Code Editors)

Step 2 — Ask ChatGPT to generate the chatbot (single HTML file)

Students will take their final, verified Q&A chain (from the FigJam → PDF → text debugging step) and ask ChatGPT to produce a single, self-contained HTML file that runs in OneCompiler (HTML, CSS, JS). Keeping everything in one file (HTML with inline `<style>` + `<script>`) prevents path/import issues and makes copy-paste testing instant.

Copy-paste starter prompt (students use this in ChatGPT)

Prompt:

“Create a single-file web chatbot that I can paste directly into OneCompiler (HTML/CSS/JS). Return one complete HTML document with inline `<style>` and `<script>` (no external files, no frameworks).

UI requirements:

- *Centered card, header with the title '[YOUR BOT TITLE]', scrollable chat area, input box, Send and Reset buttons.*
- *Pressing Enter also sends.*
- *Logic requirements:*
- *Rule-based Q&A from my list (exact answers, do not invent content).*
- *Case-insensitive matching; normalize text (trim, lowercase, strip punctuation).*
- *If no match, reply: ‘Sorry, I don’t understand that yet.’*
- *Add a short typing indicator (...) ~400ms before the bot replies.*
- *Keep last 20 messages in localStorage.*
- *Styling: clean, readable, responsive (max-width ≈ 520px, rounded corners, padding).*
- *Here is the FINAL Q&A list to implement exactly:*
- *[PASTE YOUR Q&A HERE]*
- *Return only the final HTML code block.”*

This is just an example prompt, they can change the settings and styles of the chatbot with the help of ChatGPT.

CHATGPT CODING

ChatGPT (AI Text Generator) + OneCompiler (Online Code Editors)

Step 3 — Debugging and sharing

Student instructions (what to do after they get code)

1. Copy ChatGPT's entire HTML output.
2. In OneCompiler (HTML/CSS/JS), select all and replace with that HTML.
3. Click Run.
4. If it works: great — save the link.
5. If it fails: copy the exact console error and paste back to ChatGPT with:
6. "Here's the exact OneCompiler error: [paste error text]. Please fix and return a single complete HTML file."

Once you are happy with the result of the chatbot and it actually does its job, you can share the result with other and do peer-to-peer testing.

Click Deploy



Give the Chatbot a name and number and you will get a shareable link.

Once you have the link, submit your BOT to the GoogleForm, both the **link and the CODE.**



**Co-funded by
the European Union**

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the National Agency. Neither the European Union nor National Agency can be held responsible for them