



UNLOCKING AI CHALLENGE 2026

From idea to app: **Flint + Snack**

Step-by-step guide for students: create your accounts, build with artificial intelligence in Flint and test your app in Snack with a well-organised code structure.



The School of the Future 2026 · Cognita Chile

Challenge duration: 4 days, from 23 to 26 November 2026

Teams of 3 to 5 students · Remote, asynchronous format

Before you start

In this challenge you will build a web or mobile app, simulated in Snack/Expo, that makes complex knowledge (medical or financial) accessible. You will use two tools as a team:



`app.flintk12.com`

It is the artificial intelligence that **writes the code with you**. You explain what you need and it suggests the code, helps you fix errors and explains how it works. **Its use is mandatory.**



Snack

`snack.expo.dev`

It is the **online editor and compiler** where you paste the code, run it and see your app working in the browser, in an Android emulator or on your own phone.

Your route in 5 steps

#	Step	What you will achieve
1	Create your Snack account	Save your project in the cloud and share it with your team and the jury.
2	Log in to Flint with your school email	Access your Flint account (already created) and get it ready to work.
3	Create the project in Snack	Have the app's foundation and your code organised (recommended structure or your own).
4	Flint → Snack cycle	Build the app file by file, testing each step of progress.
5	Review and submit	Working app, up-to-date logbook and Transparency Report.

⚠ Golden rule of the challenge

You can use AI freely, but **your team must be able to explain everything the AI generates**: the logic of the prompts, the design decisions and how the code works. Never paste code you do not understand without asking Flint to explain it.

Your app **cannot integrate or operate as a chatbot**: Flint is used only to build it.

1

Create your Snack account



With your Snack account you will be able to save your projects, view them from any computer and share a public link to your app.

- 1 Open `https://snack.expo.dev` in your browser (Google Chrome is recommended).
- 2 Click **Log in** (top right) and then **Sign up** to create a new account.
- 3 Complete the form: **email address** (use your school email), **username** and **password**. For the username, follow the recommendations in the box below.
- 4 Check your inbox and **confirm your email** using the link you receive.
- 5 Go back to `snack.expo.dev` and log in. You should see your username in the top-right corner.

🛡️ Your username: identifiable, but safe

Your username will be visible to other people (for example, in the public link to your app). It must allow your tutor and the jury to recognise you, **without revealing personal information**.

✓ **Use:** your first name + the initial of your surname + a number that is not your date of birth.
E.g.: `camilar27` , `tomas_g84`

✗ **Do not include:** your full surname, school name or abbreviation, year group, ID (RUT, DNI, CPF), date of birth, city or telephone number.

Optional, but highly recommended: test on your phone

- 1 On your Android phone, install the **Expo Go** app from Google Play.
- 2 Log in to Expo Go with **the same account** you created.
- 3 When you have a project open in Snack, choose the **My Device** tab in the right-hand panel and scan the QR code with Expo Go. You will see your real app on your phone.

💡 One account per student or one per team?

Each member creates their own account so that they can practise. For the official project, the team defines **a single main Snack** (saved in one member's account) and shares its link with everyone. That way you avoid having several different versions of the same app.

✅ Step 1 checklist

- ☐ I have created my Snack account and confirmed my email.
- ☐ My username is identifiable, but does not include personal or school details.
- ☐ I can log in to `snack.expo.dev` and see my username.
- ☐ (Optional) I have Expo Go installed on an Android phone.

2

Log in to Flint with your school email



Your Flint account is already created: no need to sign up, just log in with your school email.

- 1 Open `https://app.flintk12.com`.
- 2 Use your **school email**, the one that ends in your school's domain (for example `@colegiopumahue.cl` or `@americanbritish.cl`). Type it in the **School email** field and press **Next**.
- 3 Enter your **password**: it is the same one you use to access your school's institutional email.
- 4 Type a test message (for example: *"Hello, what is React Native?"*) to confirm that everything works.

⚠ School email, not the Cognita one

Do not use the @cognita.com domain or a personal email. If an access error appears, **do not create an account with a different email**: tell your teacher-tutor.

How to use Flint in this challenge

Use it to...	Example
Generate code	"Create the home screen with a title and three buttons".
Fix errors	Paste Snack's red error message and ask for the cause.
Understand the code	"Explain to me line by line what this file does".
Design the interface	"Suggest a palette and font sizes that are legible for older adults".
Simplify content	"Rewrite this medical term so that a 12-year-old can understand it".

✅ Step 2 checklist

- ☐ I logged in to Flint with my school email (not @cognita.com or a personal one).
- ☐ I sent a test message and received a reply.

3

Create the project and organise your code

A well-organised app is easier to build with AI, to fix and to explain to the jury. Before asking for code, prepare the project's "skeleton".

The structure you will see here is only a recommendation

It is an **optional example**: your team can create **its own structure** (other folders, other names). What matters is that the code is **tidy** and that you can **explain how it is organised**.

- 1 In Snack, once you are logged in, create a new project (the **New Snack** or **Create** button).
- 2 Change the project name (top left) to your team's: `UnlockingAI-TeamName`, and add a short description.
- 3 Press **Save**. Copy the Snack link and share it with your team and your tutor.
- 4 In the file panel (left), create the folders and files of your structure (you can use the recommended one or your own). To create a file inside a folder, type the full path, for example `screens/HomeScreen.js`.
Note: Snack is case-sensitive (`homeScreen` $\neq$ `HomeScreen`).

Recommended structure optional example

```
UnlockingAI-TeamName/
├── App.js           ← entry point
├── package.json    ← dependencies
├── screens/        ← one screen per file
│   ├── HomeScreen.js
│   └── ResultScreen.js
├── components/    ← reusable pieces
│   ├── PrimaryButton.js
│   └── InfoCard.js
├── data/          ← content and texts
│   └── content.js
├── theme/         ← colours and styles
│   └── colors.js
├── utils/         ← logic functions
│   └── helpers.js
└── assets/        ← images and icons
```

What is each part for?

Folder	Contains
<code>App.js</code>	Starts the app and picks the screen.
<code>screens/</code>	Each screen of the app.
<code>components/</code>	Pieces that are reused.
<code>data/</code>	Simplified texts and content.
<code>theme/</code>	Colours and styles.
<code>utils/</code>	Calculations and logic.
<code>assets/</code>	Project images.

Naming conventions suggestion

Element	Format	Example
Screens and components	PascalCase + .js	<code>HomeScreen.js</code> , <code>InfoCard.js</code>
Variables and functions	camelCase	<code>selectedTopic</code> , <code>calculateRisk()</code>
Constants	UPPERCASE	<code>MAX_QUESTIONS</code>

Repeat this cycle for **each file or improvement**. Moving forward little by little lets you spot errors as soon as they appear.

1. Ask

Tell Flint which file you need

2. Understand

Read the explanation and ask what you do not understand

3. Paste

Copy the code into the correct Snack file

4. Test

Check the preview in Web / Android

5. Record

Save the Snack and note it in the logbook

Detailed step by step

- 1 Ask for a specific file in Flint.** Tell it what your app is about, state the file path and what it should do.
- 2 Read the response before copying.** Check that the file path matches your project structure and that the code is complete (no "..." or omitted parts).
- 3 Copy the complete code** with Flint's copy button.
- 4 In Snack, open or create the file with that exact path**, select all of its content (Ctrl + A / Cmd + A) and replace it with the new code (Ctrl + V / Cmd + V).
- 5 If Flint mentioned a library**, Snack usually shows a prompt to add it; accept it or check that it appears in `package.json`.
- 6 Look at the preview** (right-hand panel): **Web**, **Android** or **My Device** tabs with Expo Go. Test on Web or Android before considering anything finished.
- 7 Red screen or error?** Copy the full error message and take it to Flint along with the file's code, and ask it to explain the cause. Do not delete code at random.
- 8 When it works, press **Save**** in Snack and note down in the logbook: prompt used, result, problems and learning.

⚠ Mistakes to avoid

- Pasting the code into the wrong file or mixing two files into one.
- Asking for "build the whole app" in one go: the result is hard to fix and to explain.
- Copying only part of the code when Flint provided the complete file.
- Several members working on the same file at the same time: coordinate who edits what.
- Writing keys, passwords or real personal data in the code.
- Asking Flint to add a chatbot or AI assistant to your app: this is not permitted.

5

Review and submit your project

The final submission is on **26 November**. Before submitting, go through this checklist together.

✓ Final checklist

- ☐ The app opens and works in **Snack (Web or Android view)** without errors.
- ☐ A person who does not know the app can use it **without prior instructions**.
- ☐ The app genuinely solves the chosen problem statement (it is not just a mock-up).
- ☐ The content is simplified, verified and written in clear language.
- ☐ The code is **organised in folders and files** with a clear structure (the recommended one or your own) that the team can explain.
- ☐ The Snack is saved, with the team's name and a working **public link**.
- ☐ The whole team can explain how each file works and the main prompts.
- ☐ The **Daily Logbook** is complete (objectives, prompts, problems and learning for each day).
- ☐ The **Transparency Report** (400–500 words or a video of max. 2 min) explains the architecture and the logic of the prompts.
- ☐ The app **does not integrate or operate through a chatbot** (Flint is used only to build it).

Remember: the logbook is also assessed (10%)

Each day, record what you asked Flint, what worked, what failed and what you learned. Copying the real prompts you used is the best evidence of conscious use of AI.