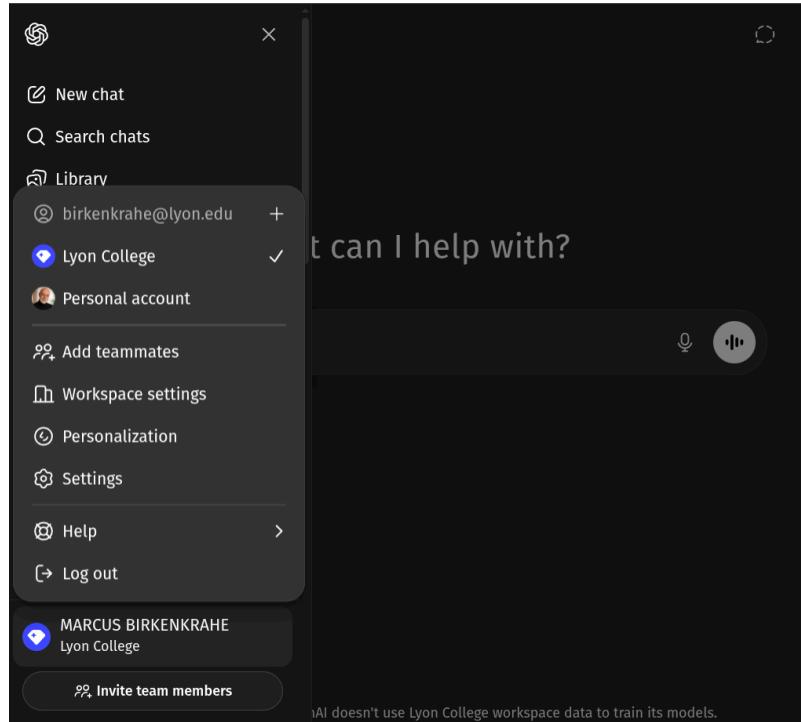


## Objectives

- [ ] Create a funny birthday card web app with AI alone
- [ ] Learn how to make the web app work on your PC
- [ ] Iteratively improve the web app
- [ ] Extract the building blocks of the process
- [ ] Understand how the app runs on your computer
- [ ] How to make additional improvements
- [ ] How to debug AI responses

## Creating an app with AI deep ai ds482

- Adapted from a course by Andrew Ng on deeplearning.ai (Jan'26).
- Demo example: a **funny birthday card web app** that runs locally in a browser (no server, no internet). This is a **cognitive agent**.
- **Prompting** means telling the AI **what you want it to do**, not how to do it.
- Code along with me:
  1. Open [chatgpt.com](https://chatgpt.com) in a browser (preferably Google Chrome).
  2. For best results, make sure that you're logged into the Lyon College team account:



3. Start a New Chat.

## Building the app in steps

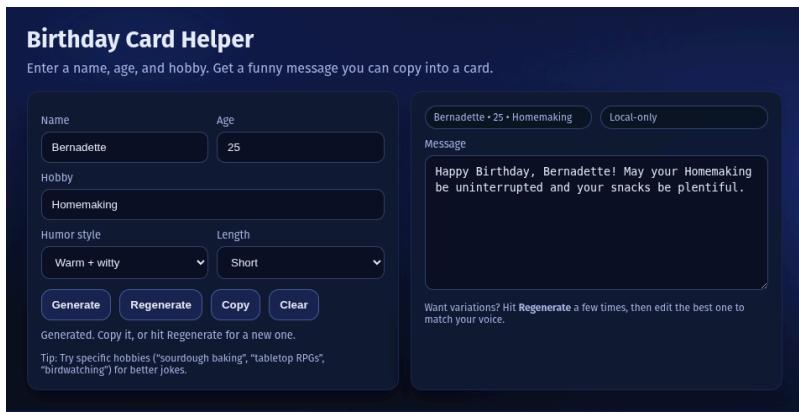
We iteratively refine the prompt and observe how the app changes.

## Minimal goal

1. Enter this prompt in the CLI:

"Create a web page to help me write birthday cards. When I give a person's name, age and hobby, it should give me back a funny message."

2. After a while, the AI will produce a window with code. Copy it to the clipboard.
3. Open an editor - Notepad on Windows, nano on Linux, and paste the clipboard content into the editor.
4. Save the file as `birthday-card-generator.html` (the `.html` is crucial).
5. Open the file explorer and find your file. Open it with the Chrome browser. You should see a working app:



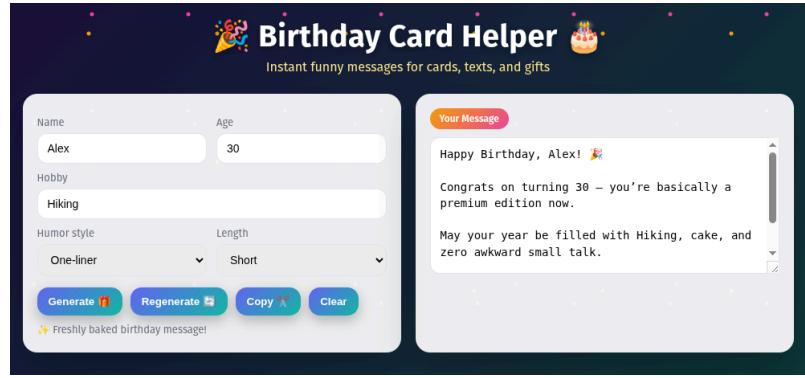
6. Use the app to make sure it works.
7. If it does not work, specify the error and ask the AI to fix it. You will get new code that you'll have to save as `.html`. Save it with a new version number, like `birthday-card-generator1.html`.
8. If everything works as you expected it to, move on.
9. **Bonus question:** Could anyone else open your web app on another PC?

## Improve appearance

1. In the same session/conversation, enter a new prompt:

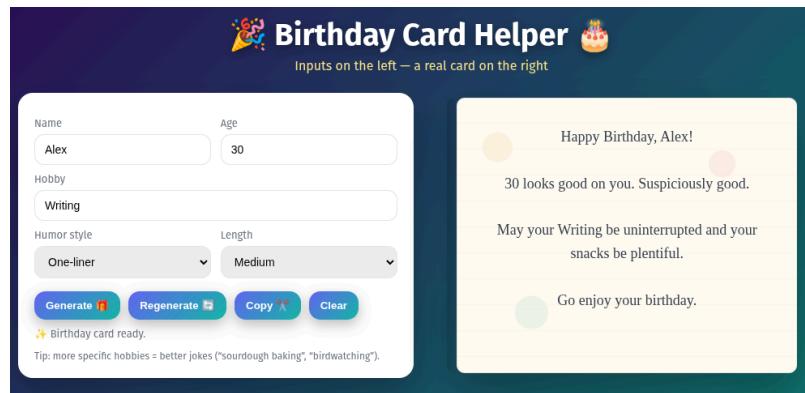
"Make it prettier by adding a festive title and colors."

2. Go through the same process as before: Save code as HTML file and render it in the browser.
3. Rinse and repeat until the web app renders in your browser.
4. This is what I got:



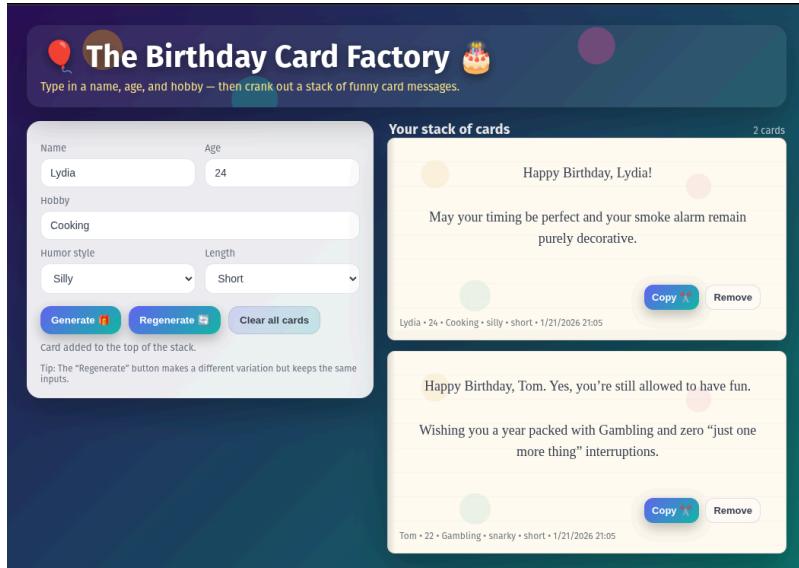
## Specify layout

1. Enter another prompt: "Display the card on the right side, and make it look like the inside of a birthday card."
2. Rinse and repeat the process to render the web app.
3. This is what I got this time:



## Add behavior

1. "Add a fun title at the top of the page. And instead of replacing the old cards, keep them visible and stack them underneath the new ones."



## Extracting the building blocks

From the evolving prompts, we can identify reusable **specification elements**:

- 1. Goal:** "Create a web page to help me write birthday cards."
- 2. Input:** "When I give a person's name, age and hobby, ..."
- 3. Output:** "... it should give me back a funny message."
- 4. Layout:** "Display the card on the right side, and make it look like the inside of a birthday card."
- 5. Features:** "Add a fun title at the top of the page. And instead of replacing the old cards, keep them visible and stack them underneath the new ones."

## Test: all blocks in a single prompt

What happens if we give the AI **all five elements at once**?

(Open a **new conversation** to avoid context effects.)

Create a web page to help me write birthday cards.

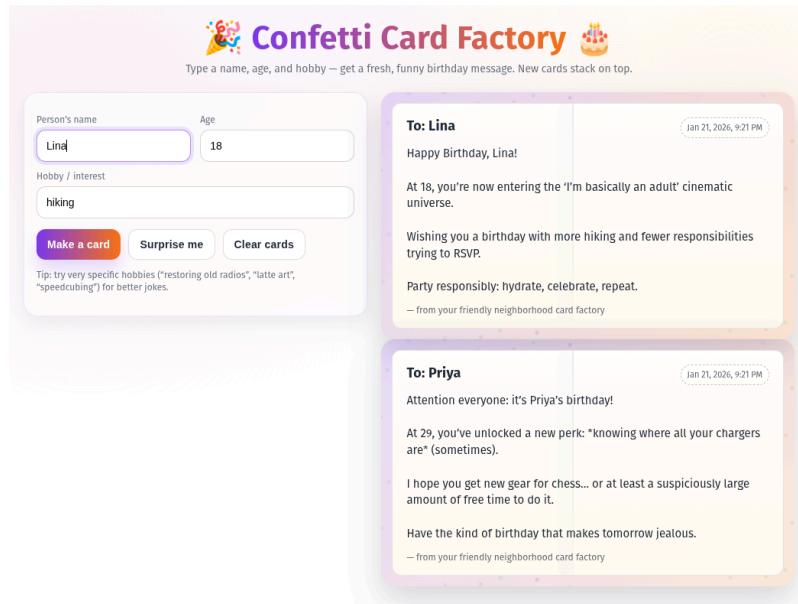
When I give a person's name, age and hobby, it should give me back a funny message.

Use a festive title and colors.

Display the card on the right side, and make it look like the inside of a birthday card.

Add a fun title at the top of the page. And instead of replacing the old cards, keep them visible and stack them underneath the new ones.

This is what I got: I got a "surprise me" button without asking!



## Observations

- Always start with a **goal**.
- You do not need to use all blocks.
- The order of blocks usually does **not** matter.
- Even with precise prompts, results will vary.

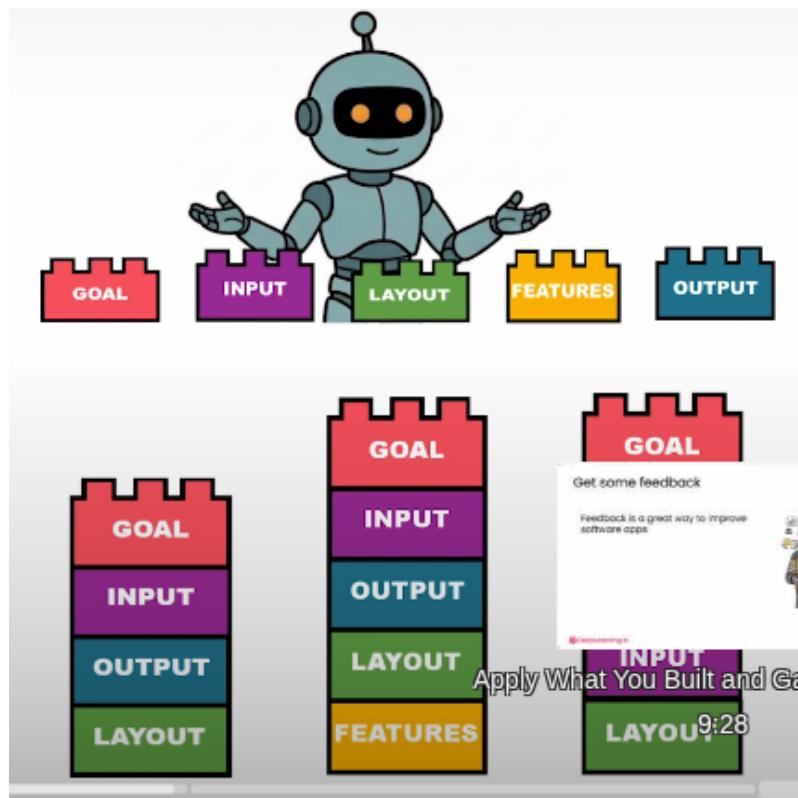


Figure 1: Source: Andrew Ng, deeplearning.ai

- Vague goals (e.g. "Create a web page to help me write birthday cards.") often lead to surprising and inconsistent results.

## How does this run on your computer?

An HTML file is plain text whose structure the browser understands.

- HTML controls layout
- CSS controls appearance
- JavaScript controls behavior (interactivity)

No server and no internet connection are required.

## Improving the app (exercise ideas)

- Add five clearly labeled inputs: Name, Age, Hobby, Adjective, Plural Noun
- Add an "I'm feeling lucky" button that fills inputs with random, predefined funny words
- Add buttons to copy cards to the clipboard
- Make titles and subtitles more decorative
- Change the color theme (e.g. blue)

## Debugging with AI

- Example prompt:

"Nothing happens when I click on the 'Generate Card' button. Can you fix it for me?"
- You do **not** need to understand all technical details to ask useful debugging questions.

## Final note

- Feedback from others is one of the best ways to improve software.
- You can also ask the AI for ideas ("brainstorm") — just remember that the ideas are often generic and need human judgment.

Author: Marcus Birkenkrahe (pledged)

Created: 2026-01-31 Sat 16:25