Hello! In the remaining time (approx 1 hour), we'd like you to create a simple webapp, following the specs below. If you know the technologies, please use PHP (ideally Laravel) + Nuxt. If not, use whatever you are most comfortable with.

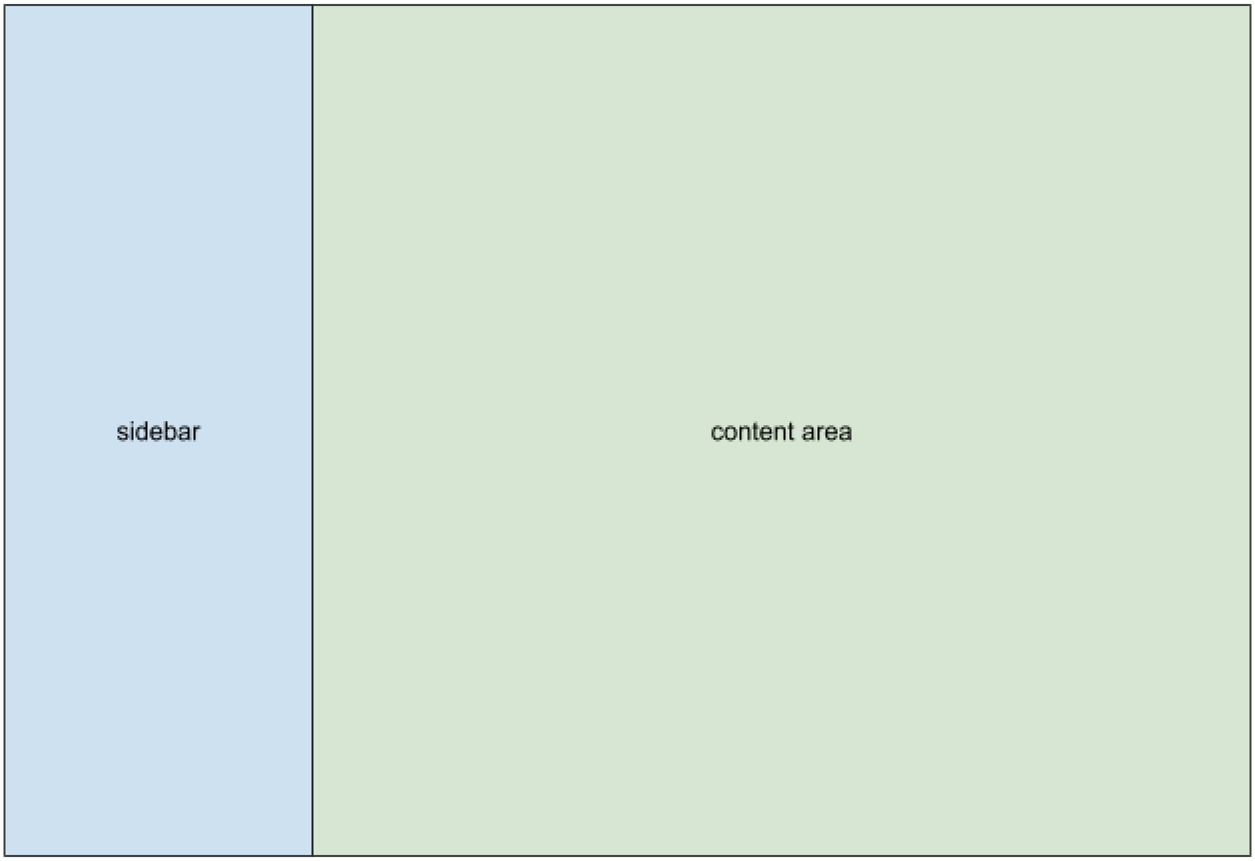
You can use Stack Overflow, ChatGPT/Claude/copilot, or whatever you like to help you move fast. In fact, we'd encourage you to. Just show us how you are interacting with these tools.

You don't need to finish all of this during the allotted time - just see how far you get, and then submit your code to us.

Step 1 - Dashboard

Set up your project. Create an API and frontend. When you navigate to /, serve the default page.

It should have two panes, like this:



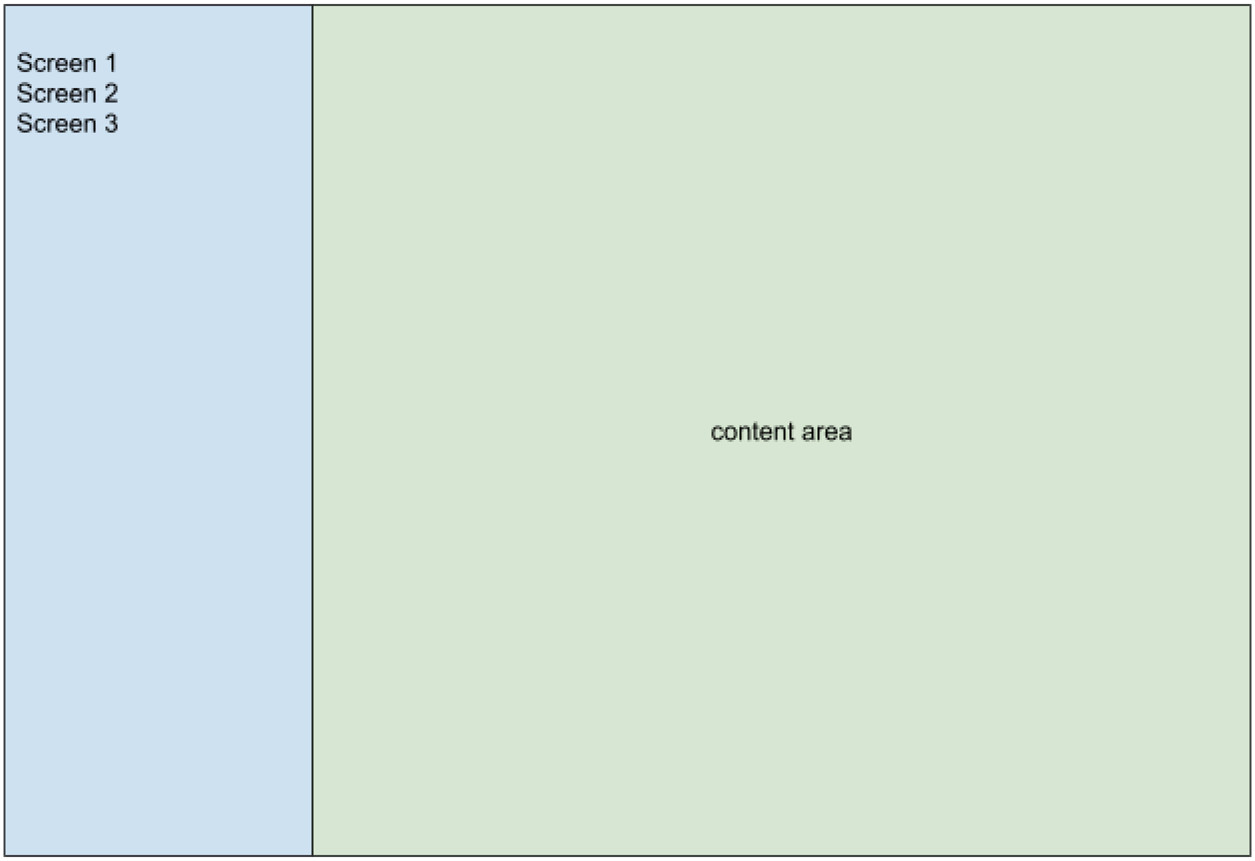
Step 2 - navigation links

Make 3 links in the sidebar as shown. When clicked:

a) Make a call to the backend (either for the entire rendered template, or for json data to populate the frontend template, depending on how you are building the app)

b) Render the result in the content area

    i) For now, the content of the three screens can just be "This is screen 1", "This is screen 2", "This is screen 3"

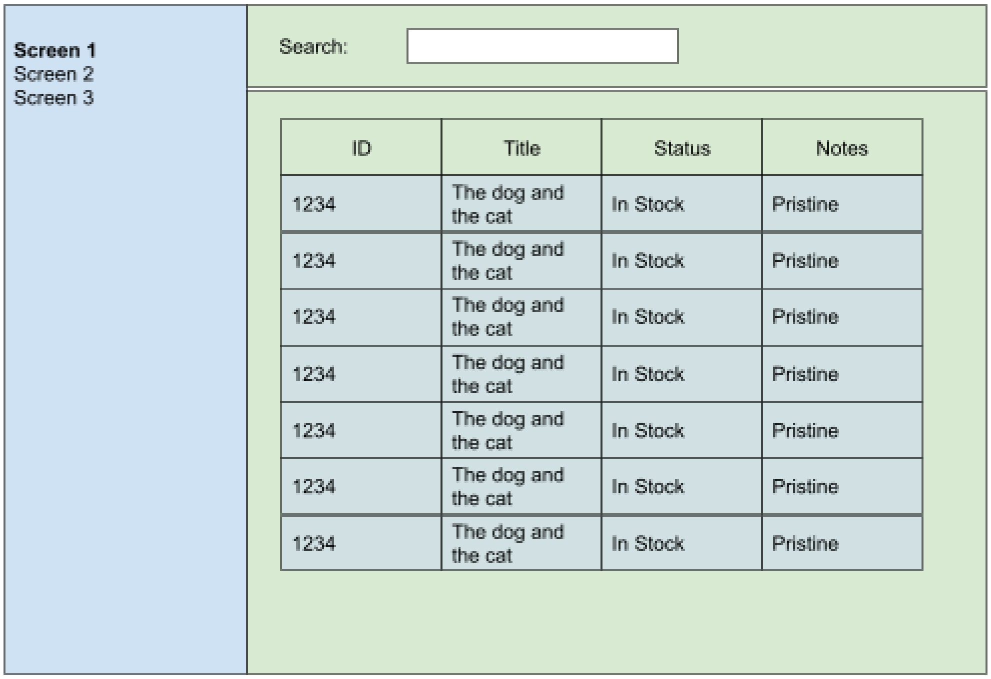


Step 3 - screen 1

Now we're going to build out each of these screens.

On screen 1:

a) There's a search bar at the top. It fires 300ms after the last key-up event, and resets if the user keeps typing. Enter should also submit. If search is emptied then get 1 page of records.
b) By default when we load screen 1, we should return 1 page of records (say, 20 records, but not important). These must be served from the backend.
c) Search should hit the backend. Search the id, title and notes fields; return matching results.

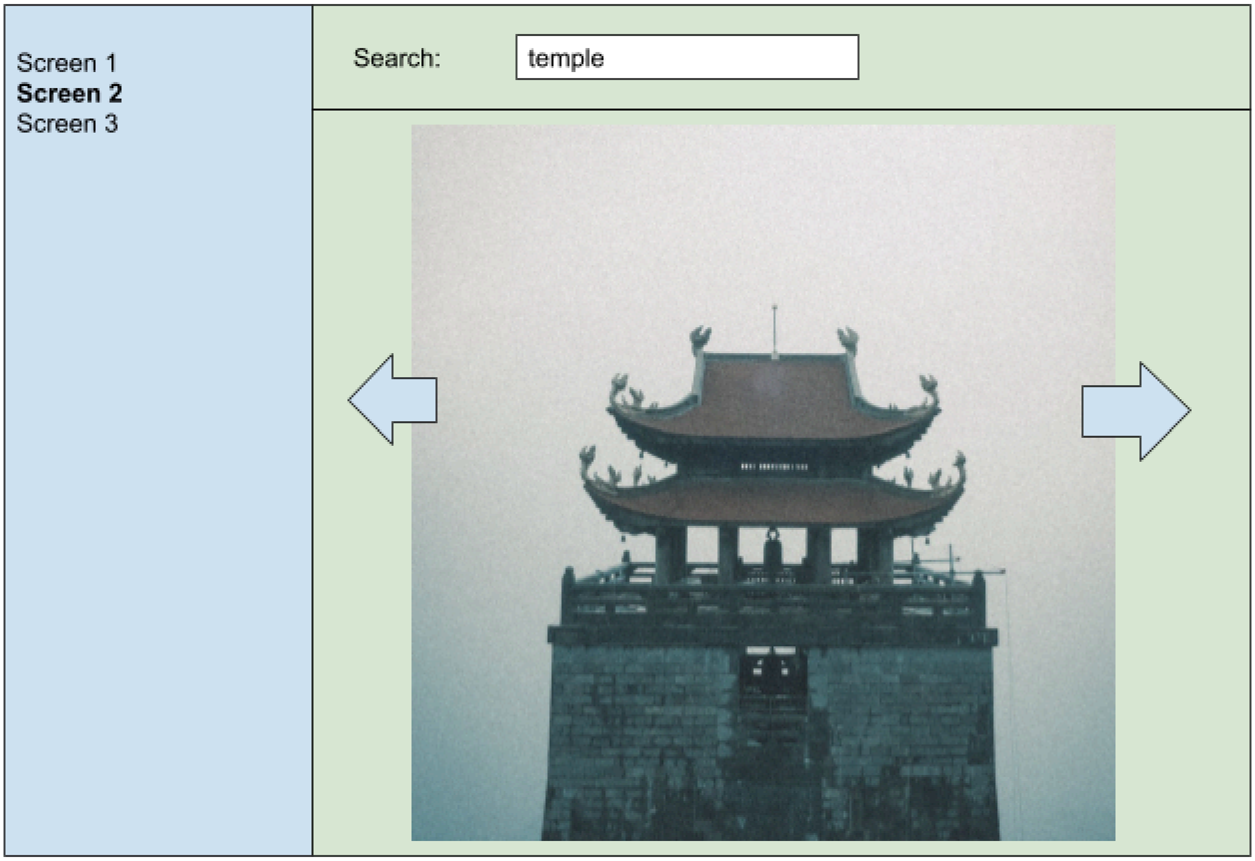


ID	Title	Status	Notes
1234	The dog and the cat	In Stock	Pristine
1234	The dog and the cat	In Stock	Pristine
1234	The dog and the cat	In Stock	Pristine
1234	The dog and the cat	In Stock	Pristine
1234	The dog and the cat	In Stock	Pristine
1234	The dog and the cat	In Stock	Pristine
1234	The dog and the cat	In Stock	Pristine

etc. make up your own stubbed data. ChatGPT is pretty good at generating records if you give it an example.

Step 4 - screen 2

- Now we're going to integrate a third party API. This screen is an image search that lets users browse forward and back between matches.
- Sign up to unsplash.com/developers (free account) and get an API key.
  - Note that you need to confirm the registration email
  - After that, go to https://unsplash.com/oauth/applications and click New Application, in order to get your access key
  - Use this access key in order to interact with their search API
    - Auth: https://unsplash.com/documentation#public-authentication
    - Search: https://unsplash.com/documentation#search-photos
- For this search, the enter key submits. We don't want to spam pexels.
- The arrows switch the image by paging forward and back using the pexels API.

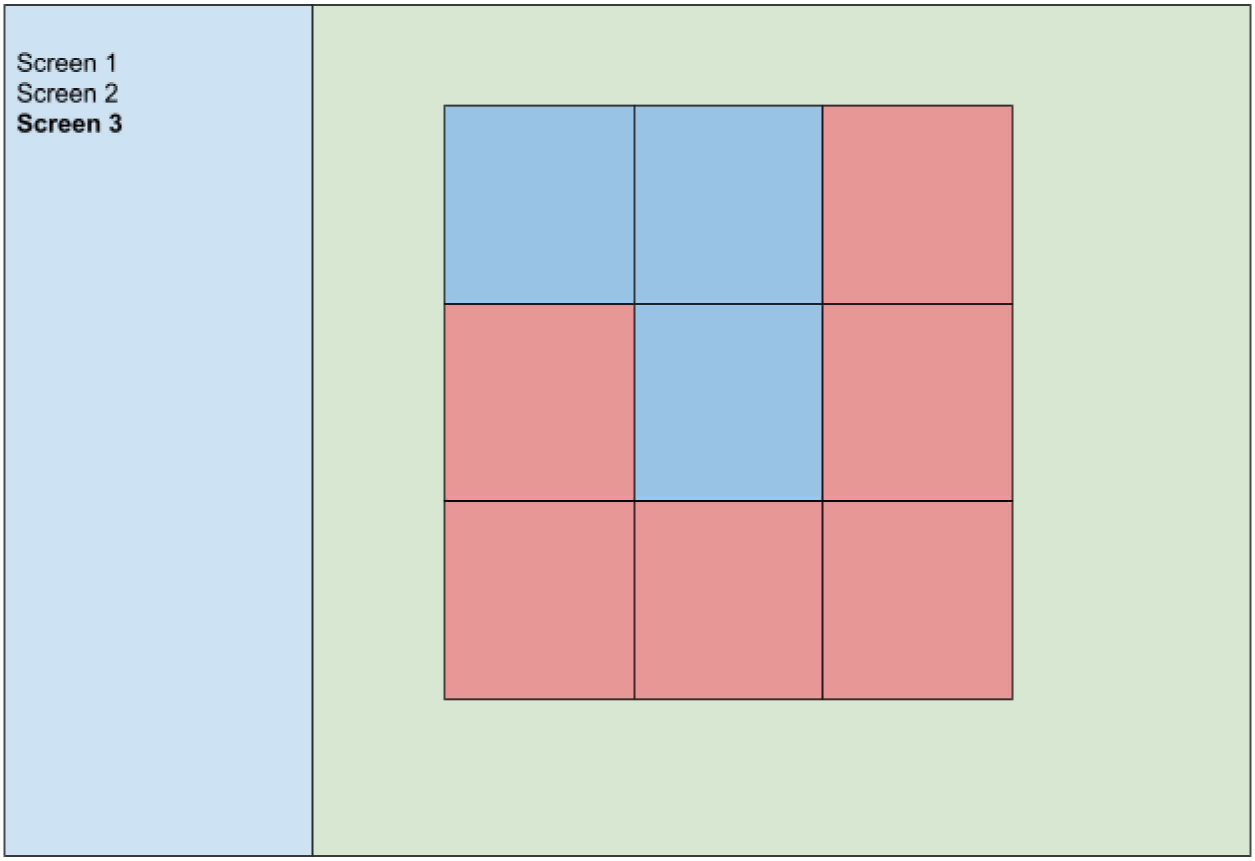


Search: temple

Step 5 - screen 3

Great, you're still going! Good effort. We'll make this one fun!

- On the page we show a 3x3 grid of squares, colored either red or blue.
- Query the backend api to get the initial state, which can be randomized.
- Clicking a blue square turns all of the red squares that it is touching (either up/down or left/right) blue.
- Clicking a red square turns all of the blue squares that it is touching red.
- Either way, the square itself toggles to the opposite color.
- In all cases, make a call to the backend to inform it which square was clicked on, have the backend do the calculations and return the new state to the frontend.



Step 6 - you still have time?

Rockstar. Spend the remainder of your time making the application better.

Improve styling to make it look prettier.
Refactor code to improve the organization.
Add tests.

Whatever appeals to you - no choice here is wrong. Just go polish, and impress us!