Pelago

Hi there,

Welcome to the engineering challenge! With this challenge, we want to get an idea about how you approach engineering problems and how do you translate your solution into code.
We would like to give you a problem that you would likely come across when working with the team.

We know you have a day job and asking you to solve such a task is asking a lot. Remember, it is not about showing us how well you can polish your solution. Instead we want to get an idea about your real-life code.

Our process of reviewing your code is straightforward: two engineers will sit together, timebox the review to 30 minutes to **run the code** and **review your approach to the problem.**

We will then rate your implementation based on **completeness**, **readability**, **maintainability** and **performance**. We will then share our view on your assignment no matter whether we choose to continue with your application or not. We are sure there is something to learn from this task - for you and for us!

This is also an opportunity for you to say you do not like such problems - fair enough, we can understand this domain is not for everyone. Interviewing for a job, in our view, is for both sides to learn enough about the other to be able to make a decision. In case you decide not to continue, we would nevertheless kindly ask you for feedback on our process, so we can improve and make it a pleasant experience for all applicants.

Your submission will remain confidential to Pelago and will not be shared externally. It will only be viewed internally by Pelago employees who are directly involved in the interview process and hiring decisions.

Pelago

# Front End Interview Challenge

The highly successful ( and fictional ) movie review website *heycinema* has called upon you to help build its new search page.

Their data is retrieved from the **Open Movie Database (OMDB) API**. You will need to go to http://www.omdbapi.com/apikey.aspx, request an API key and read the documentation for their search endpoint before using it.

As a developing business with a strong brand *heycinema* has attached an excerpt from their style guide on the next page. These colours and design patterns should inform your page layout and design.

Use your expertise to create a user-friendly page using **ReactJS** with the following:

1. The basic *heycinema* header, as seen in the style guide.
2. A **search bar** that takes a string and calls the OMDB search endpoint on submit.
3. A list of **search results**, displayed as **tiles** with the following information:
    - Image
    - Name
    - Year of release
4. Loading and error states handled.

Your app should be mobile friendly taking extensibility, maintainability and performance into consideration.

Please include a **README** for the following information:

- Steps to **build, run and test** the code.
- **Information on any architectural decisions** eg. choice of tech stack, code structure, approach towards styling, state management. ( This is the place where you get to justify your rationale for using that crazy cool tech you want everyone to know about)

If you could not or did not want to finish something, this is also where you can tell us why and how you would have continued.

Please submit a **link** to your hosted app ( using Surge, Heroku, AWS etc ) as well as a link to a **public** git repository ( Github, Gitlab, BitBucket etc ).

## Stretch

Want to go the extra mile ? You are free to add anything you think will improve the codebase or user experience. Here are some ideas:

- Paginate your search data or implement infinite scrolling
- Perform searches asynchronously.
- Add micro interactions and animations to delight your users.
- Create an advanced search that allows users to refine their results further.
- Add unit and integration tests.

# heycinema styleguide excerpt

## Typography

**Headings: Helvetica Neue bold 18px**
Body: Helvetica Neue regular 16px

## Colours

Primary
#7FAAE4

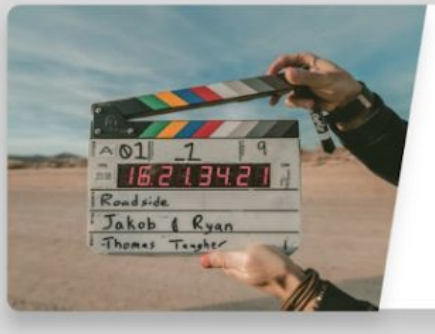
Secondary
#834DB7

Tertiary
#34283F

## Header

hey
cinema

## Cards




**Heading**
Subtitle

Callout

## Input

*Label*