Movies search application

Time: 1 hr

Requirements (Part-1)

- Fetch the list of movies using below mentioned API.
- Each movies should be rendered in a "Card" like component which will encapsulate following movie related data -
  - Image
  - title
  - Genre
  - IMDB rating
  - Release Date
- Every movie card should also have heart-shaped(❤) "like-button". This button will fill/unfill red colour based on click toggles.
- All the movie cards should be rendered in 3 * 3 grid.
- Add a search bar (without search button) at the top of this grid.
- Search bar should reactively filter the movies below using search term against the movie-title.

Movies API specs:

- API verb: GET
- Return type: JSON list of movies
- Endpoint: https://hexanovate-1oc3v5uf6-thephenom1708.vercel.app/api/movies

Requirements (Part-2) : Implement these requirements only after

finishing part-1.

- Add a radio button filter below search-bar with following options:
  - All Movies: (default-selection): Render complete list of movies
  - Favourites: When selected render only "liked" movies
- Debounce the search with 2 seconds time delay.
- Maintain the state of this radio selection in the route. This should enable back-button navigation. e.g. If user has selected "All Movies" and then selects "Favourites" then after clicking browser back-button, user should go back to prior "All Movies" selection.
- Maintain state of search term in route.

Notes

- Complete assignment should be coded in ReactJS/NextJS.
- Candidate is free to style the UI elements as per his/her creativity. But all the styling is expected to be implemented using CSS/SASS and candidate should not use any other styling library/framework.
- We do not expect very fancy styling. Basic design is enough. Implementation of functionality is more important.
- Candidate has to share a link to codebase once the assignment is complete. Hence it is advisable that candidate should either push codebase to any version control system or use sharable online code editors (e.g. codesandbox)