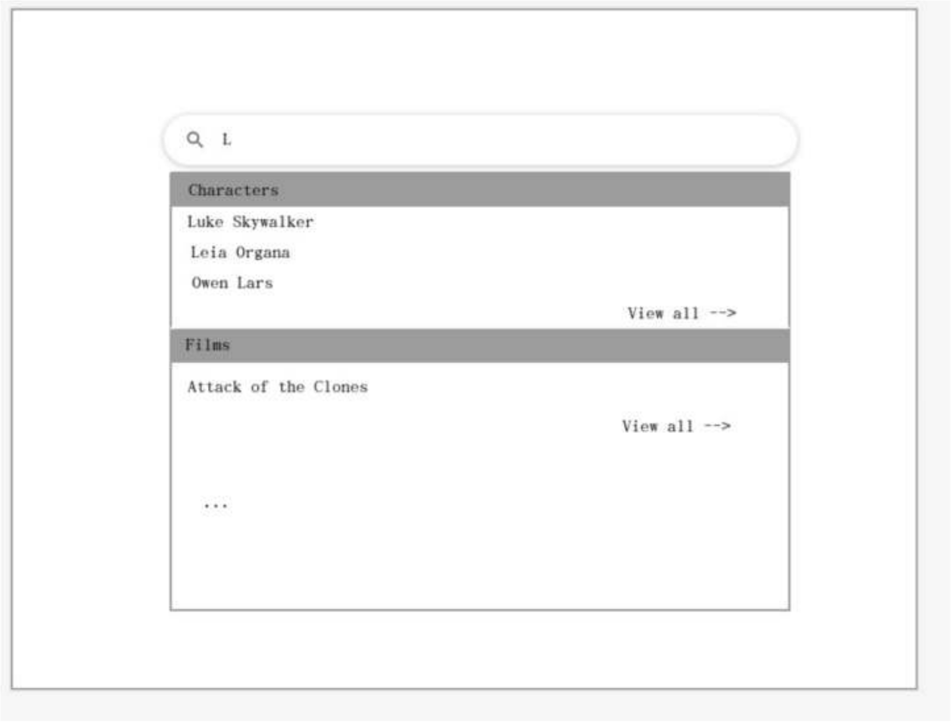
Your assignment is to implement a small web application based on StarWars API (https://swapi.dev/). The app will consist of 3 pages: Search, List. Form.

Note: Please read the whole exercise before starting

Search Page.



The page should present a search field.

When the user starts typing, the search term should be searched in all entities of the API (films, people, planets, etc. - as described in the API's docs). For example, the URL for searching “L” in “people” is https://swapi.dev/api/people?search=L. allow a 200 ms debounce between typings.
The results should appear in a popup from the bottom of the search field (auto-complete style), divided to categories accordingly, showing a maximum of top 3 matches for each category.
At the bottom of each category, there should be a button with the caption “View All”, leading to the category's page.

Expectations:

1. the main requirements (the must have)
    a. correct layout the positioning of the components
    b. it works as expected
    c. Click on view all for Characters routes to List page, for others convery user that it’s not implemented.
2. Bonus Points
    a. Highlight the searched term in the results
    b. Api request cancellation for on-flight requests upon new search term.
    c. using Suspense.
    d. error boundary.
    e. optimizing package size.

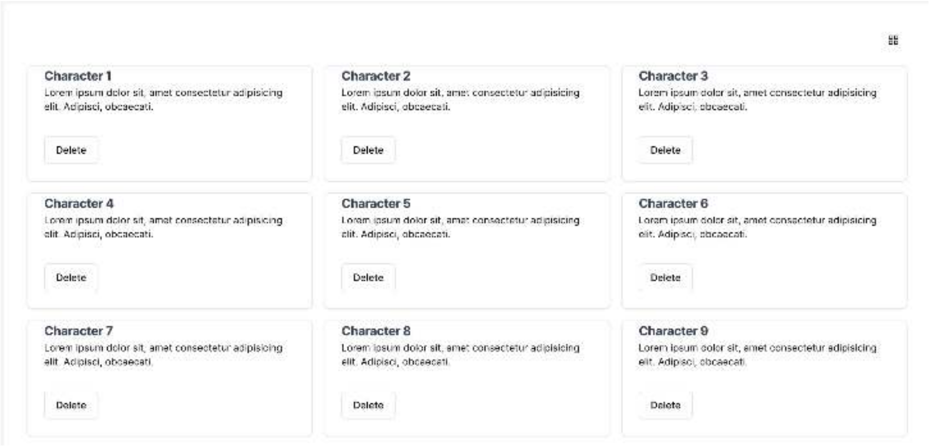
List Page for Characters.

A click on “View All” for characters should lead to the list page. You are required to implement only one list page, the Characters list (as shown in the screenshot – sorting and paging not mandatory).

Note: The UI is just an example, feel free to add your ideas. It doesn;t have to be pretty. But it needs to be functional

By default the UI should show a List view. The User will have an option to change to grid view. You can use a toggle switch, checkbox, button-group or anything that is your preference to let the user change the layout from list to grid and vice versa.

Same item-card used for both grid view and list view. For list view the contents are organized horizontally and in the grid view the contents are organized vertically.



Character 1
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.
Delete

Character 2
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.
Delete

Character 3
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.
Delete

Character 4
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.
Delete

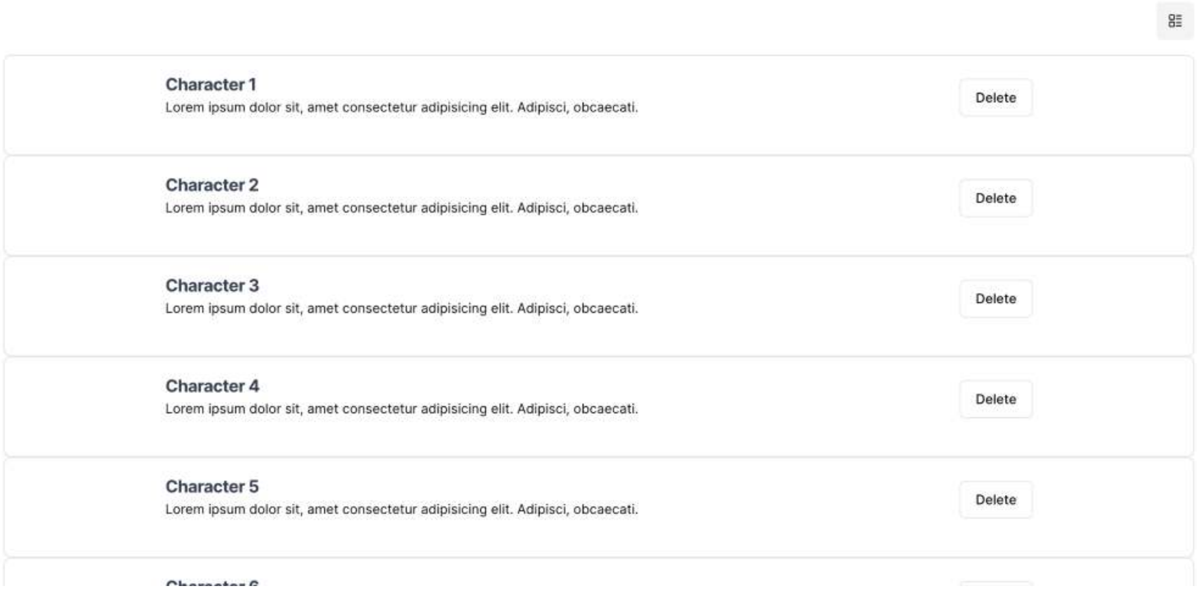
Character 5
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.
Delete

Character 6
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.
Delete

Character 7
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.
Delete

Character 8
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.
Delete

Character 9
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.
Delete



Character 1
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.

Character 2
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.

Character 3
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.

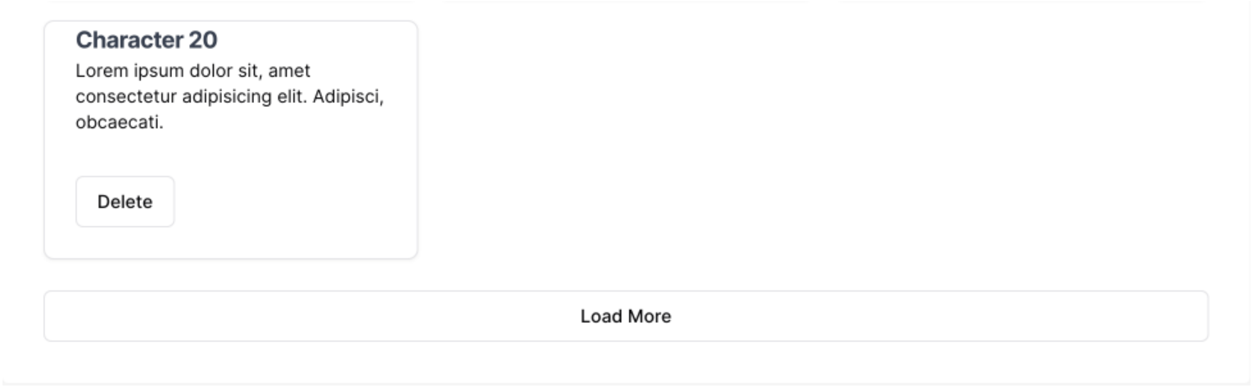
Character 4
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.

Character 5
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.

Character 6

Pagination:

Initially load 20 items. A “load more” button in both list and grid view to load more contents from the API.



Character 20
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Adipisci, obcaecati.

Delete:
Clicking “delete” will delete the current character (since API doesn’t have a DELETE API, delete only from local state).

Edit:

Click “edit” will open an edit form in a dialog for the current character (edit only the name, gender (male, female, unknown) and birth year, update in state only).

Expectations:

1. the main requirements (the must have)
    1. Functionality should work as mentioned above
    2. functional Load more button
    3. switch between grid view and list view
    4. support at least medium and large screens (min 780px ~ max 1440px)

2. Bonus points
    1. build a responsive UI with mobile first approach.
    2. support small, medium and large screens.
    3. use only css for the layout and responsive design. (No framework) suggesting css grids