# Enhance Home Component for Pagination Support

## Description:

Enhance the Home component to support pagination when displaying movies. Integrate the Pagination component, call the getMovies service each time the user clicks on a new page by changing a new state to track pagination. The structure of this new state should be { filters: { pagination: { currentPage: number, totalPages: number } } }.

This task ensures that the Home component is enhanced to support pagination, providing a more dynamic and user-friendly experience when navigating through movie data. The integration of he Pagination component allows users to easily switch between pages, triggering the fetching of data for the selected page.

**///////////////// query Params Version**

Enhance the Home component to support pagination when displaying movies. Integrate the Pagination component, call the getMovies service each time the queryParams in the url change, you will need to to use the hook “useQuery” to be able to detect the changes in the URL an the set the correct state

You need to add this new state { filters: { pagination: { currentPage: number, totalPages: number } }

Each time the user clicks on the page button you need to navigate to the new URL.

Your URL should looks like this “/?currentPage=X”

NOTE: totalPages should update when you have the answer from the API

## Acceptance Criteria:

1. State Definition:
   1. Define a new state named filter using the useState hook.
   2. The structure of the filter state should be { filters: { pagination: { currentPage: number, totalPages: number } } }.
2. Pagination Component Integration:
   1. Integrate the Pagination component within the Home component, passing the necessary props such as onSelectPage, currentPage, and totalPages.
3. Pagination State Update:
   1. Update the filter state each time the user clicks on a new page using the onSelectPage callback.
   2. Query Params Version: update the queryParm currentPage state each time the user clicks on a new page using the onSelectPage callback.
   3. When the URL change you need to update the filter state
4. Fetch Movie Data on Page Change:
   1. Modify the getMovies function call in the Home component to include the current page from the filter state.
5. Unit Testing:
   1. Write unit tests for the updated Home component, covering scenarios such as changing pages, updating the filter state, and ensuring that the getMovies service is called with the correct parameters.

## Definition of Done:

1. The filter state is defined within the Home component with the structure { filters: { pagination: { currentPage: number, totalPages: number } } }.
2. The Pagination component is integrated within the Home component, passing the necessary props.
3. The filter state is appropriately updated each time the user clicks on a new page.
4. The getMovies function in the Home component is modified to include the current page from the filter state.
5. Unit tests for the updated Home component pass successfully, covering various pagination scenarios.