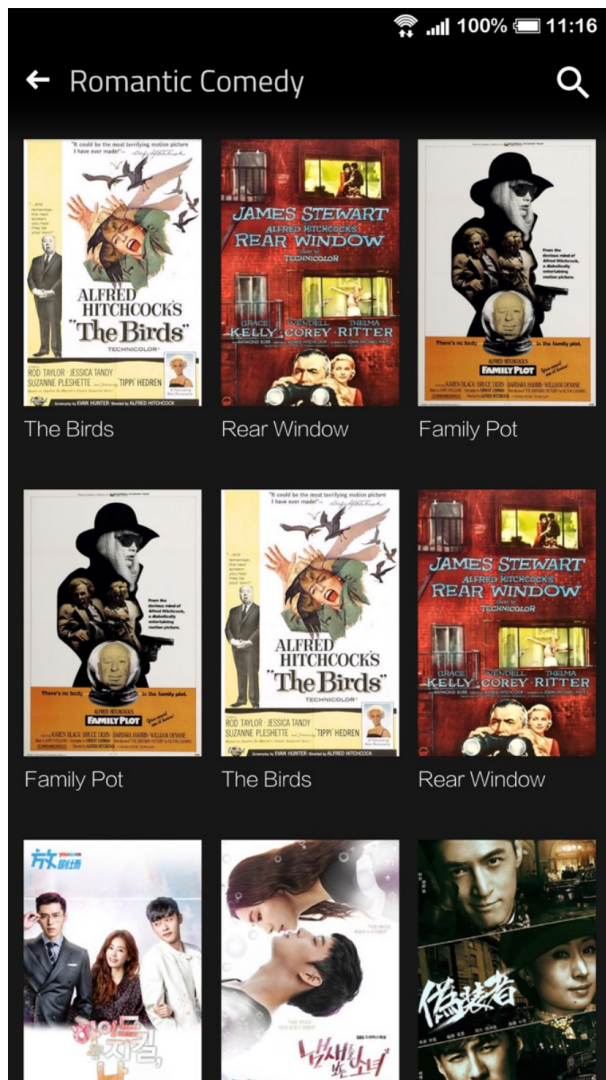


# Requirements

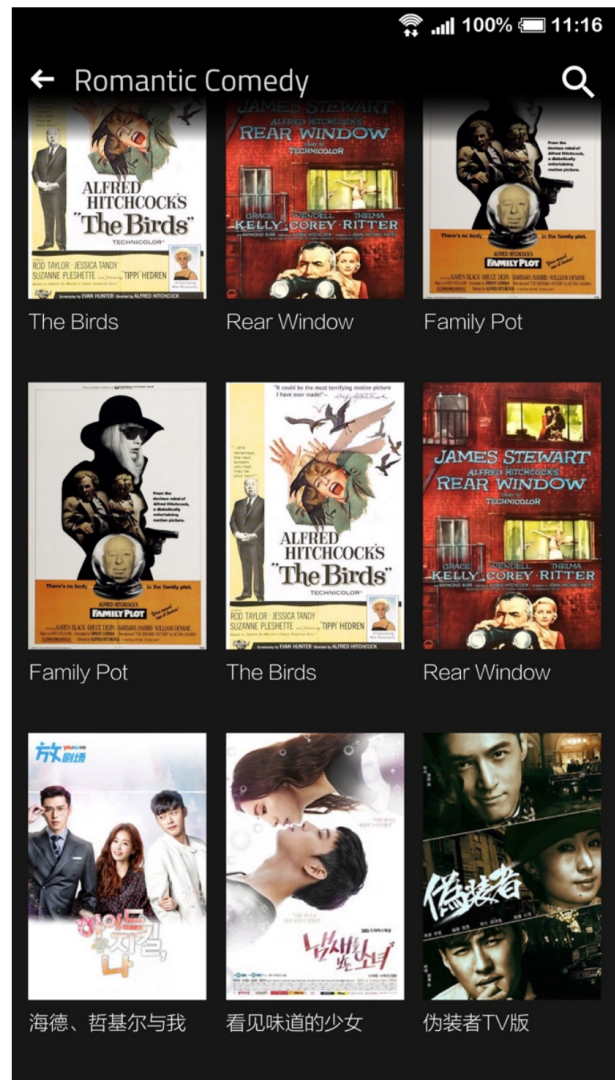
Develop the mobile browser version of Content Listing page as per the design given in the Design Requirements section below, using ReactJS. You can use any prebuilt react UI component / frameworks that you think are going to help speed up the development process.

## Design Requirements

1. Your output must closely align with the design provided below.
2. Implementation is required only for the mobile portrait view, and there is no need to consider desktop or landscape dimensions.
3. The Grid component should always show three columns, even paddings, ensuring responsive behavior on various mobile resolutions.
4. There should be no visible scroll bars.
5. In cases where specific instructions are not provided in the design, make informed decisions on placement and sizing.



Initial View



Scrolled View

- Global Font : **Titillium Web** (<https://fonts.google.com/specimen/Titillium+Web>)
- Thumbnail Aspect Ratio : **2:3**
- Background Color : **#171717**
- Text Color : **#FFFFFF**

## Functional Requirements

1. The listing grid should have vertical scrolling enabled while restricting horizontal scrolling.
2. App must implement lazy loading of contents - image grid must load gradually as the user scrolls down. Data for the first page can be loaded during app load.
3. Client Side Search must be implemented by filtering the data that is already loaded in the UI (*not by calling any API*). Search results must be rendered on to the main view itself without refreshing/reloading the page or taking the user to a new page

## Technical Details

- Base API URL : <https://test.create.diagnal.com/>
- Data API Pages : </data/> [Example: <https://test.create.diagnal.com/data/page1.json>]
- Images API : </images/> [Example: <https://test.create.diagnal.com/images/poster1.jpg>]

1. The Images API has all the images that should be used for this test. This includes poster images which are referenced to in the Data API discussed below and the UI images like back button, search button etc. You can access a list of all the images from the following URL: <https://test.create.diagnal.com/index.html>
2. The Data API has a few JSON files with naming **page(#NUM).json**. These are to be used as mock data for the grid (including specifying titles and poster images to be displayed). You will need to retrieve the data, one JSON page at a time as the user scrolls down in the Grid. Do not fetch all JSON pages at once, they should be loaded as the user navigates and approaches the end of the current data set. This lazy loading should be done in a seamless way without the user noticing that additional data is loading (no pausing of scroll or blocking loaders during scroll).
3. Two content items on Page 3 have edge cases which you will need to think creatively on how to solve without breaking the UI consistency.