Application must have 3 screens that the user will see. Actual number of widgets can be more than 3 of course. Application will combine 2 architectures: Bloc, MVVM. But for the user it must behave in exactly the same way. The app can have components that are shared within these 2 architectures.

Screens:

1) Screen has 2 buttons Bloc, MVVM. Each button will open the Screen2 in a specified architecture. The screen is in portrait mode.
2) Screen has a list of movies. The data for the list is hardcoded. On tap it opens Screen3. The screen is in portrait mode.
3) Screen has a detail for the selected movie. Image that is downloaded by URL. The screen is in portrait mode.

Requirements:

1) Data is hardcoded on the app data layer.
2) Images for the screens must be taken by URLs and cached (no not be downloaded each time)
3) Screen 2 and 3 should support Bloc and MVVM architectures, depending on user selection on the Screen1.
4) Both (Bloc and MVVM) variants behave in the same way.
5) Tap on Screen 2 opens Screen 3 with a proper details for the selected movie.
6) Back on screen 3 opens Screen 2 on the same scroll position as it was before navigating to the Screen 3;
7) Back on screen 2 opens Screen 1.
8) Application supports rotation on Screen 2 and 3
9) In landscape mode both 2 and 3 are combined on a single screen. ⅓ part on the left takes a list of movies, the rest on the right is for the movie details. In this doc the screen is called a Landscape Screen.
10) App supports back navigation. In the title a back button is shown. For the landscape screen the back leads to screen 1.
11) Landscape screen should have list items highlighted once it’s tapped.
12) Screen 2 in portrait should not have an item highlighted
13) Once screen 2 is rotated to Landscape, the Landscape screen is shown with no item highlight and no details on the right.
14) Once the screen 3 is rotated to Landscape, the Landscape screen is shown with a highlighted item that corresponds to the data within the Detail section. And on the right the details are shown the same as on the Screen3 before rotation.
15) Once LandscapeScreen is rotated to Portrait, the Screen 2 should be shown with no item highlights.

Estimate: 2-3 weeks.

Additional task once developer finishes it earlier than in 3 weeks:
Task1.
Screen 2 should support Pull To Refresh, where the hardcoded data will be shuffled randomly on the Data layer and brought to UI.

Task2.
Create a real networking where the previously hardcoded data will be returned from an echo server, similar to https://putsreq.com/ . In case of Pull to refresh, the behaviour from Task1 should be reverted, but additional request should be sent.