Programming

Build a simple Media App VOD (Video On-Demand) application that keeps track of a list of videos the user has watched. You may use this API for a video list https://accedo-ps-programming-exam.s3-ap-southeast-1.amazonaws.com/movies.json

Application Features

1. The application should display a list of videos in a horizontal carousel on the home page, which should be scrollable. DONE
2. The user should be able to select a video to play in full screen. DONE
3. When the video is finished, the application should go back to the previous page. Done, video will exit full sreen and back to main page
4. The application should be navigable by both the mouse and keyboard, i.e. navigate using the arrow keys and enter to select. Done, carousel can control by mouse and keyboard
5. The user should be able to see a list of videos they have previously watched. Done, there are a "watch again" list at the bottom of the page
6. The application UI must be responsive to changes in the browser window / screen sizes. You do not need to implement a mobile view but it should at least adjust based on the desktop browser width. done, All are responsive

Advanced Add-on Features

- Adding lazy load functionality in your carousel Done
- Implementing multiple carousels in categories Done
- Implementing movie information popup window Done, "more info" button
- Anything else you can think of Added header, inline player, etc

Requirements

- The app should work in the latest versions of the most popular browsers (Chrome / Firefox etc.). Done

- The app should be developed entirely in HTML / JavaScript / SASS, as a **single page application** (https://en.wikipedia.org/wiki/Single-page_application) Done
- The viewing history should be saved on a server and persisted across browsing sessions. That is, if the user refreshes the browser the history should be automatically loaded. User authentication is not necessary. Done, using redux presist
- The server component should be developed in NodeJS / MongoDB. Nodejs and reactjs
- Data transferred between the server and client side should be in JSON format. JSON format only
- Detailed instructions about how to set up the app and server are required. A live demo hosted on a publicly-accessible server is highly regarded; for example AWS, Heroku, OpenShift. I have include a setup procedure
- If you are using any cloud solutions to host or develop your application, credential(s) should be provided. using OpenShift trial
- The app should be controllable by the arrow keys (to navigate) and enter (to select). Mouse support is also required. Done
- Functioning unit tests should be provided with the source code (Mocha / Chai / Jest). Done
- The app should follow the design template illustrated on the next page. Suggested tools (but not required):
    - Chrome Developer Tools (for debugging) yes
    - Frameworks such as React, Express, Angular, Bootstrap, Sass yes
    - Build tools such as Grunt, Gulp, Webpack Build by react-scripts

Extra Credits

- 100% unit test coverage Done
- Comment the code nicely Done
- Make it beautiful
- Provide a Docker image Done, checked in in Github

Delivery

Your implementation of the Media App VOD application should preferably be provided in a private Github repository (or in a compressed archive), containing

all necessary files and resources. Also, instructions on how to install and start the app are expected, preferably in a ReadMe file. Done

Note

All work needs to be original. By submitting this programming test, you are declaring that all code in the application is your original work.

Useful Resources

Playback Icons:

https://fortawesome.github.io/Font-Awesome/icons/

Movies API:

https://accedo-ps-programming-exam.s3-ap-southeast-1.amazonaws.com/movies.json

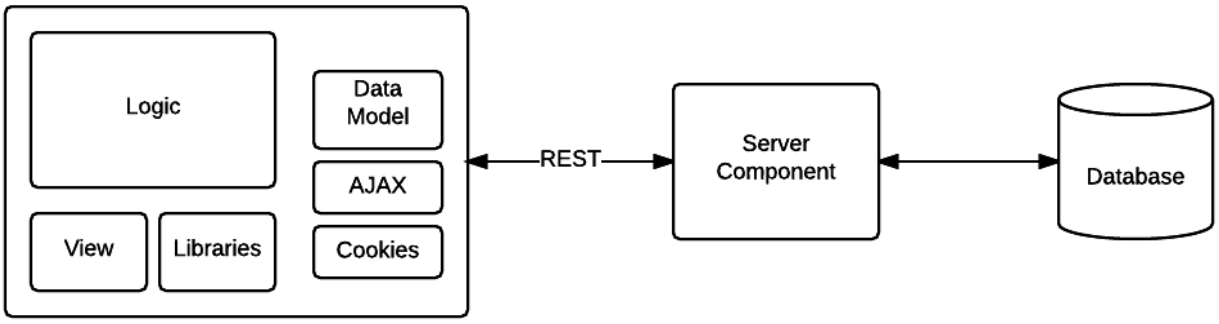
The Movie DB:

- Reference: https://developers.themoviedb.org/3/getting-started/introduction
- Movie List API: https://api.themoviedb.org/3/genre/movie/list?api_key=<api_key>&language=en-US
- Movie Category API: https://api.themoviedb.org/3/genre/<category_id>/movies?api_key=<api_ key>&language=en-US&include_adult=false&sort_by=created_at.asc

OpenShift:

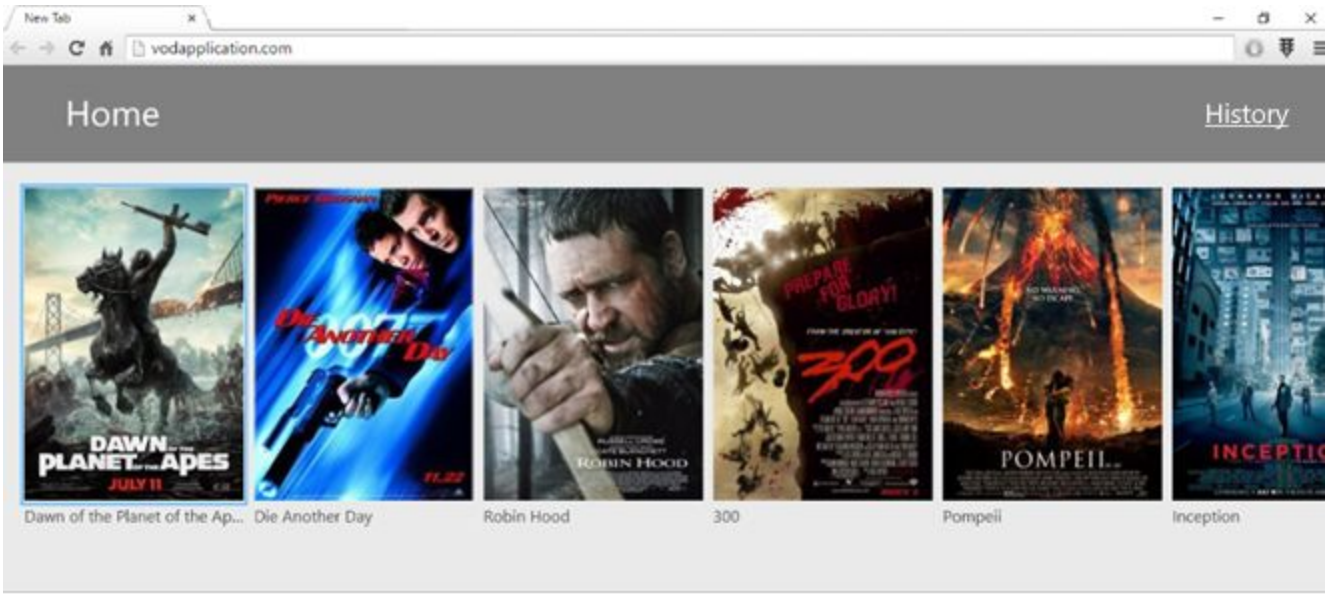
https://www.openshift.com/

Suggested Structure

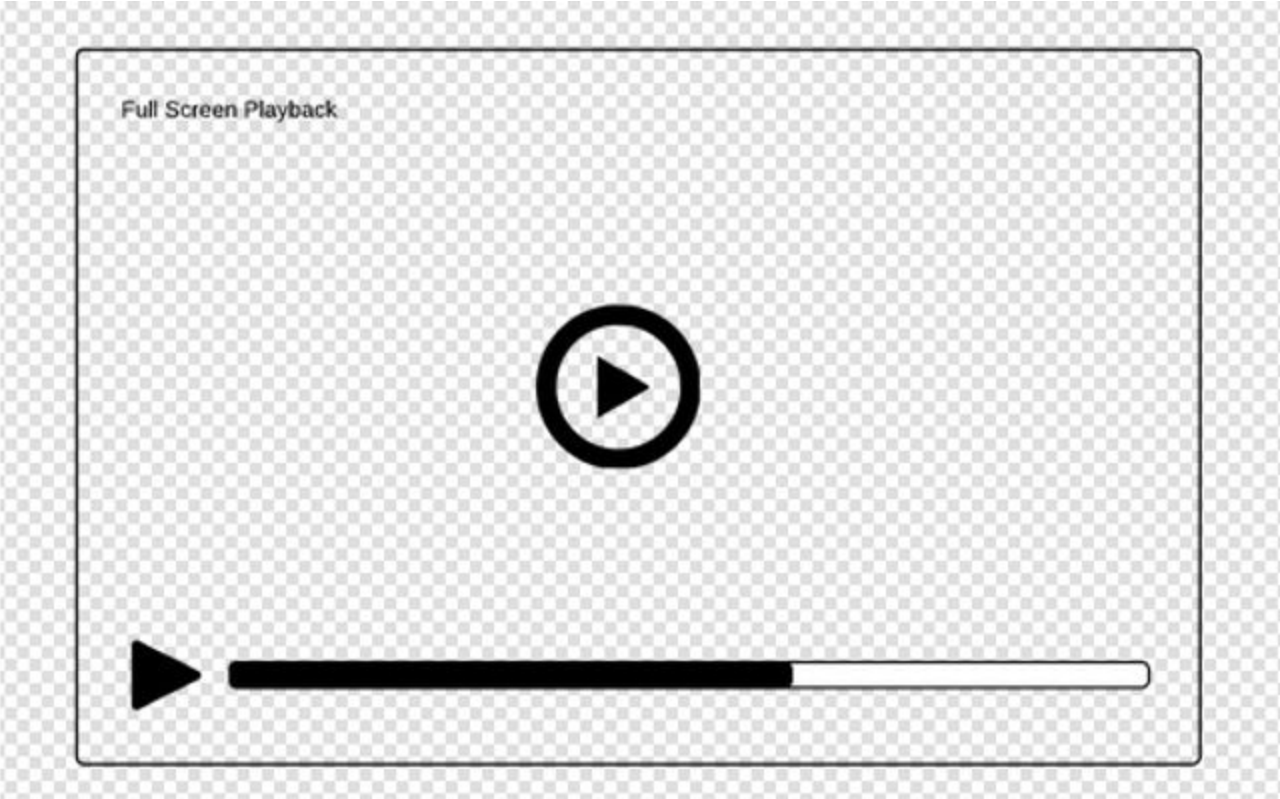


UI

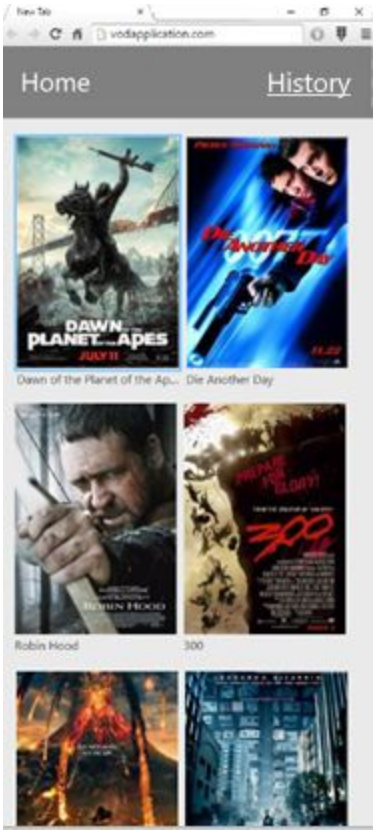
Landscape:



Player:



Portrait (Optional):



Marking Scheme (Practical Test)

Points out of 10

- Meet the requirements (2)
- Proper use of well-known frameworks and/or modern technology (1)
- Documentation (1)
- Coding Style (3)
- Deployment / setup / hosting (1)
- Creativity (Advanced Add-on features / Extra Credits) (2)

Design Question

At Accedo, our developers are involved throughout the entire product development process; from pre-sales all the way to go-live. Imagine you're a developer on a VOD project that allows users to save the videos they want to watch later.

1. Describe the data you would capture as part of this service.
2. How would make this service more efficient?
3. Once the feature is complete, how would you know that it's ready to go live?
4. How would you determine if this feature is successful?

Marking Scheme (Written Test)

Points out of 10

- Answering the questions (2)
- Show in-depth knowledge (4)
- Show industry relevance (3)
- Show creativity (1)

Design Questions

At Accedo, our developers are involved throughout the entire product development process; from pre-sales all the way to go-live. Imagine you're a developer on a VOD project that allows users to save the videos they want to watch later.

1. Describe the data you would capture as part of this service.
2. How would make this service more efficient?
3. Once the feature is complete, how would you know that it's ready to go live?
4. How would you determine if this feature is successful?

Marking Scheme (Written Test)

Points out of 10

- Answering the questions (2)
- Show in-depth knowledge (4)
- Show industry relevance (3)
- Show creativity (1)