React Code Assignment

Position: ReactJS Developer

Introduction

This document describes the design and functionality of a test project as code assignment for the position of ReactJS Web Developer. This test is designed to have a better overview of your programming skills.

Good luck!!

Description

The test project will be a ReactJS web application. The result of your work should be a working application for Google Chrome v.49 or higher.

Instructions

Create an web application for browsing the popular movies, display a short detail page of a selected movie and play a short movie trailer.

For this purpose you have to use **The Movie Data Base** API. We will use the **v3.**

API documentation:
Overview: [https://www.themoviedb.org/documentation/api]
Getting started: [https://developers.themoviedb.org/3/getting-started/introduction]

For that, you will need to register yourself and get an **api_key** [https://www.themoviedb.org/account/signup]

1. The first screen of your application should display a group of carousels with the most **popular movies**, most **popular tv series**, and the genres **family** and **documentary** [see wireframe 1]. From the response use **poster_path** value to get the poster image, and the **title** value to display the title. An example of a valid image url path [http://image.tmdb.org/t/p/w342/rqAHkvXldb9tHlnbQDwOzRi0yVD.jpg]

2. After selecting one of the assets in one of the carousels, the application should navigate to the detail page of the selected movie or series [see wireframe 2].

3. After pressing the “Watch Movie” button the application should display a full screen movie player and should automatically start the playback. Instead of using **The Movie Data Base** trailer results for that,

you will neet to use a hardcoded hls video stream [https://bitdash-a.akamaihd.net/content/sintel/hls/playlist.m3u8] together with **shaka player** [https://github.com/google/shaka-player]. Integration with **shaka player** is required for this test.

4. *OPTIONAL TASKS:*
   (I) Make the app responsive [wireframe 4]
   (II) Implement search functionality
   (III) *C*reate at least one unit test!

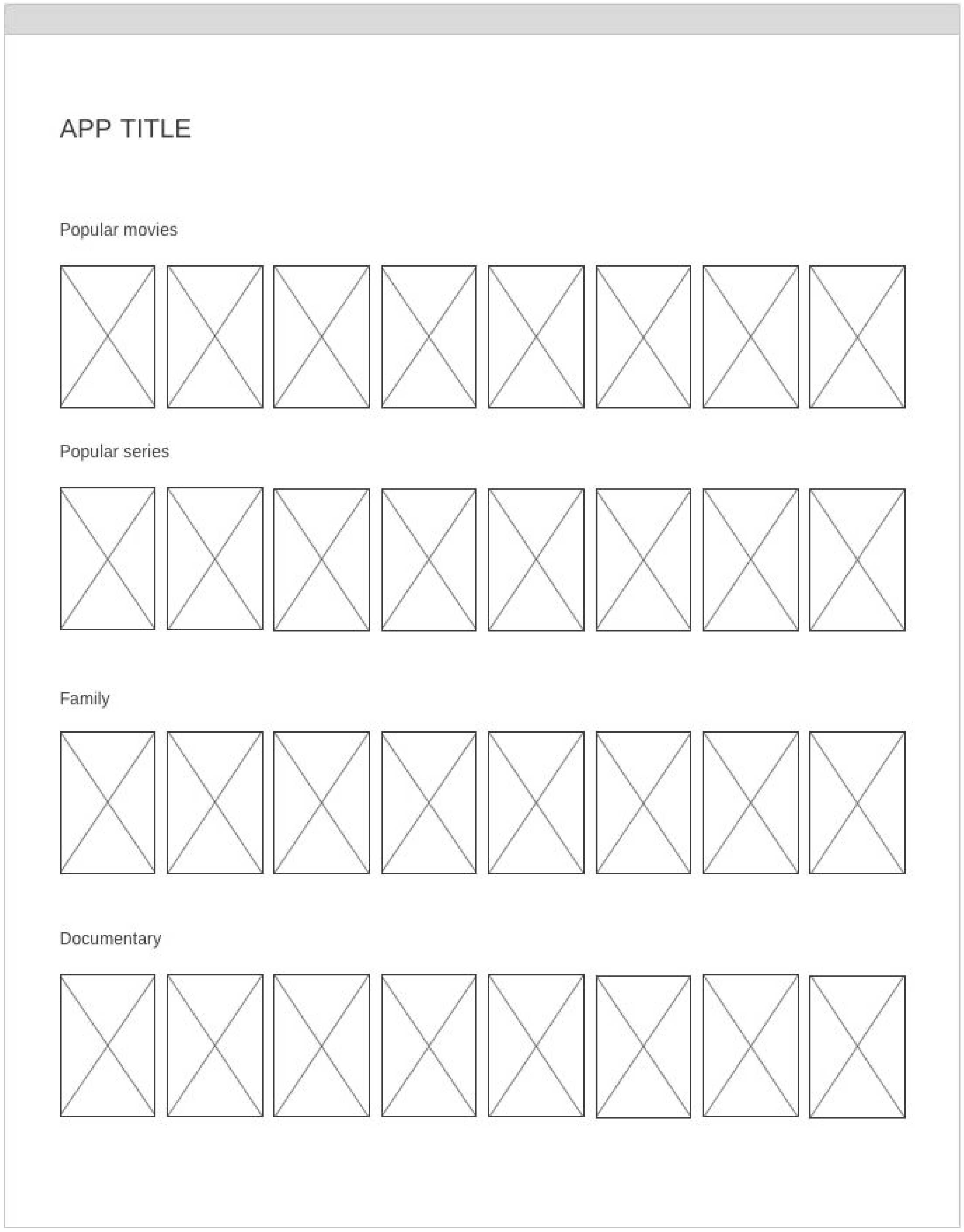
Hints

- Use the attached wireframes as a style guide & possible solution. If you think some other flow will improve the ux, go for it.
- Handle as many error cases as you can.
- A proper folder scaffolding will be a plus.
- Documentation.
- Optional tasks 1 and 2 are strongly recommended!
- Comment your code
- It's recommended to have zero warnings in the project.
- Write your code as production grade.
- Use as many libraries or utils you need. But you MUST use React.

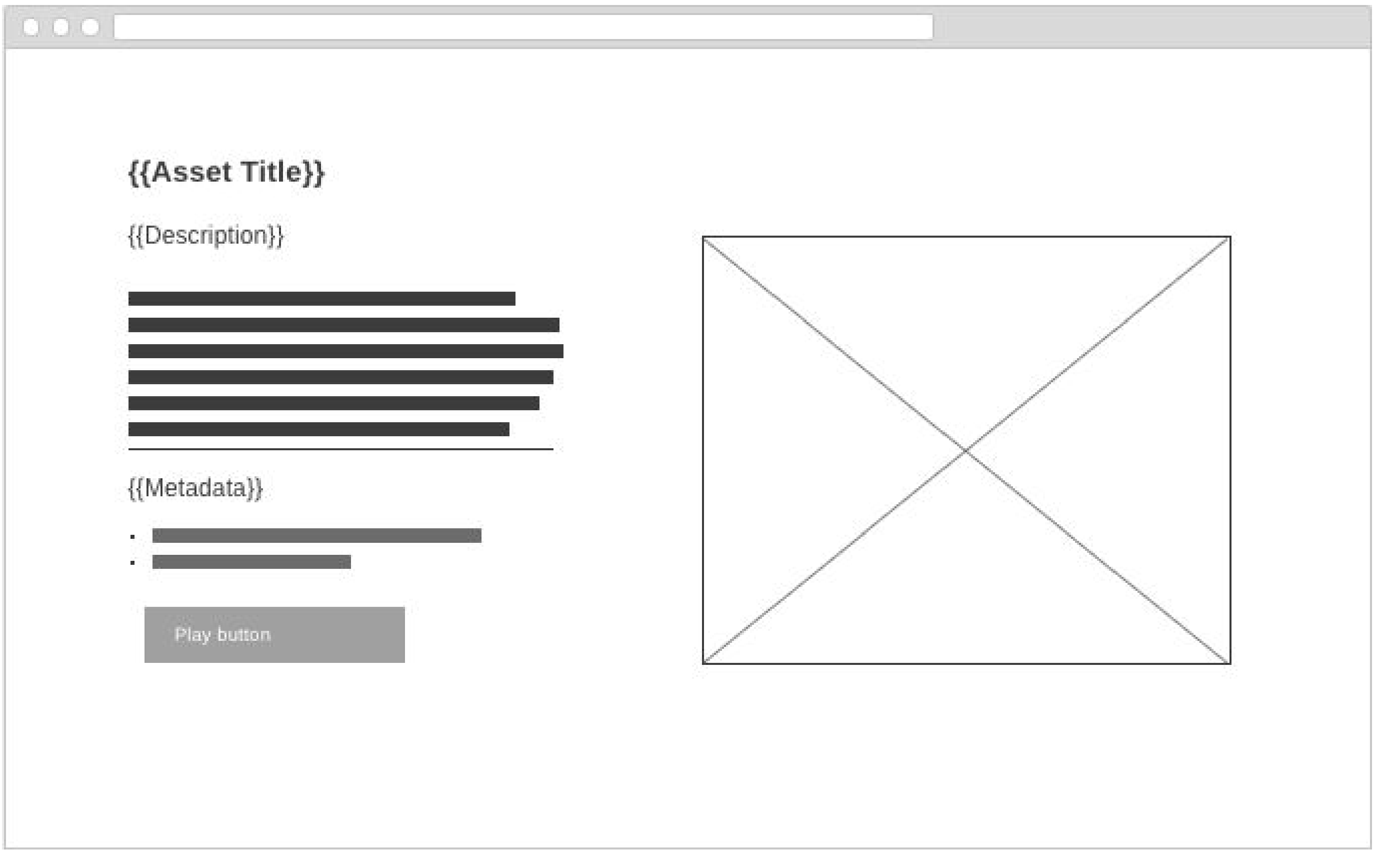
Results

The results of your work should be a working application in Google. We strongly recommend you to create a Git repository for your project, and share the repository URL with us.
Your source code will be evaluated with a possibility of personal discussion with the development team.

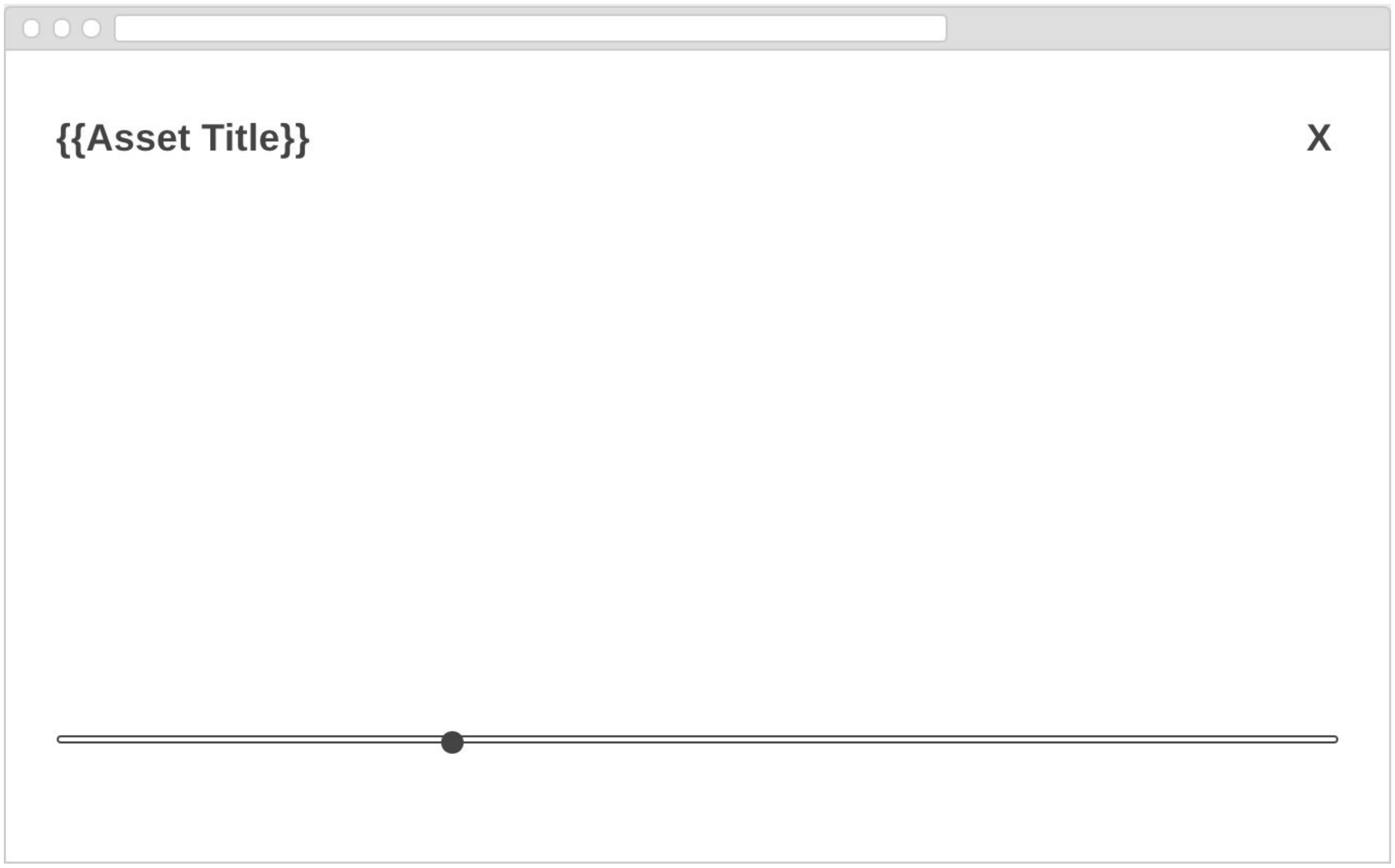
Wireframes



Wireframe 1 (Dashboard view)

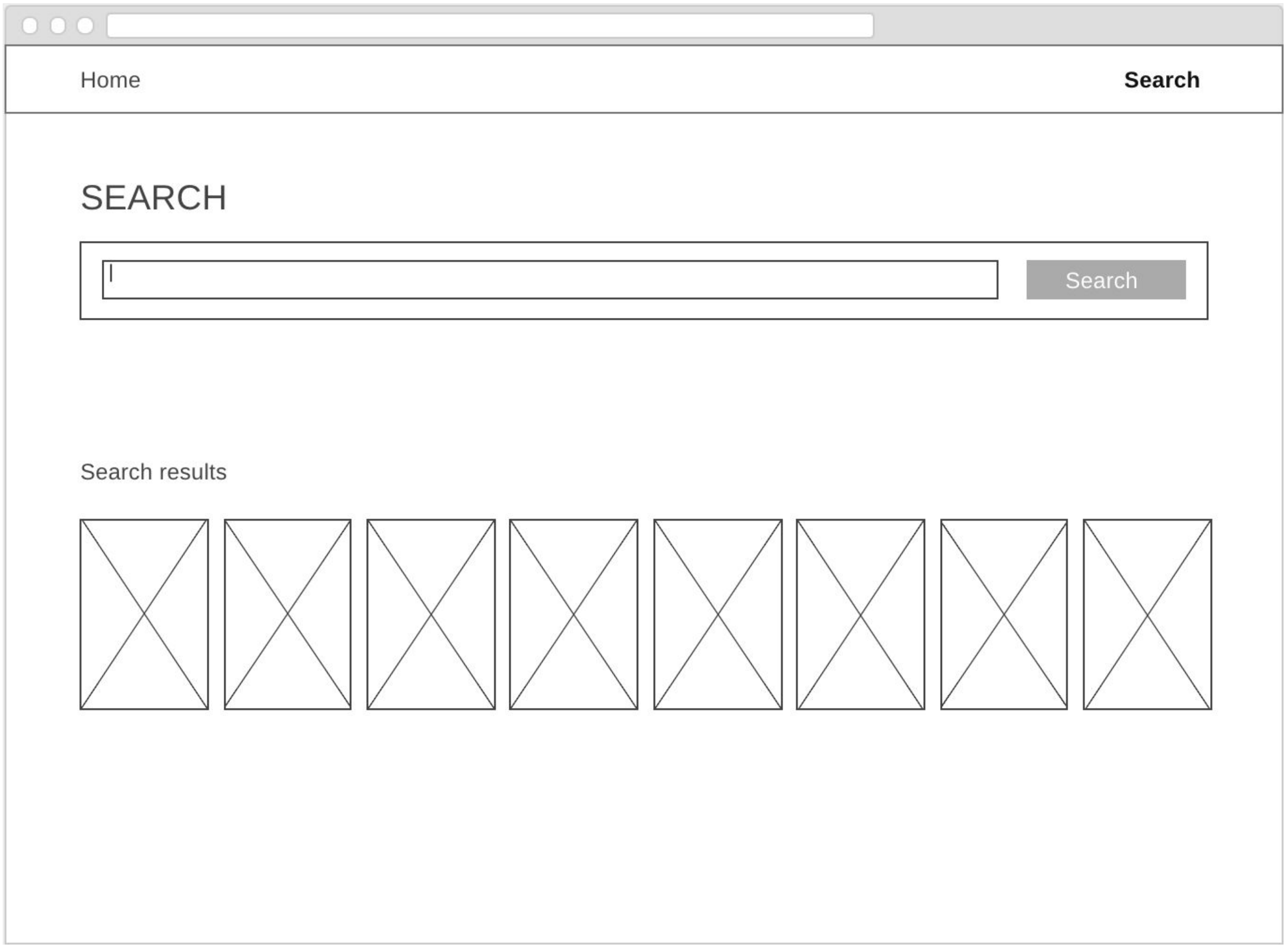


Wireframe 2 (Detail view)



Wireframe 3 (Player view)

Optional Wireframes



Wireframe 4 (Search view)