Process Book
Boxoffice 360

Team Details

Project Repository: https://github.com/anirajk/boxoffice360
Walkthrough Video:

Aniraj Kesavan
aniraj@cs.utah.edu
u0996550

Ashwini Janamatti
ashwinijanamatti@gmail.com
u0996548

Process Book – Boxoffice 360
Aniraj Kesavan – u0996550
Ashwini Janamatti - u0996548

Overview and Motivation

By the end of the 1880s, the introduction of lengths of celluloid photographic film and the invention of motion picture cameras, which could photograph an indefinitely long rapid sequence of images using only one lens, allowed several minutes of action to be captured and stored on a single compact reel of film. Some early films were made to be viewed by one person at a time through a "peep show" device such as the Kinetoscope. Movies have grown
from paltry beginnings to complex multi-year projects producing highest fidelity pictures using
3-D Stereoscopic projection etc.

There is a staggering amount of movies made over generations and there is no good visualization tool that exists today which captures the essence of the evolution of movies, nor does a solution exist which can visualize movie data per year with the ability to sort and filter based on attributes of a movie. Solutions such as IMDB exists, but they are better suited to extract more information than general exploration

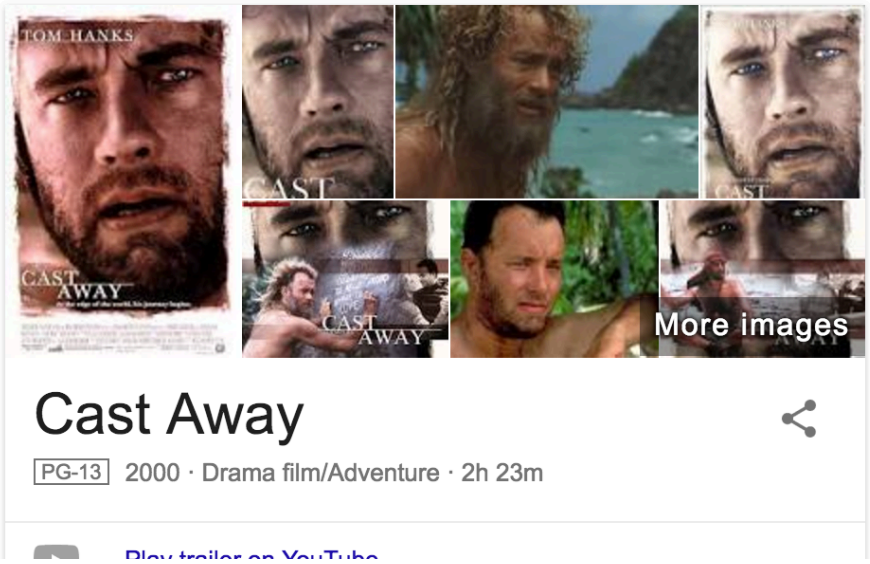
Process Book – Boxoffice 360
Aniraj Kesavan – u0996550
Ashwini Janamatti - u0996548

In this project, we plan to visualize a well curated dataset of over 5000 movies. We pre-processed the data using imdb's apis to enhance information and visualize both the holistic evolution of movies as well as enhanced views of movies released every year.

Process Book – Boxoffice 360
Aniraj Kesavan – u0996550
Ashwini Janamatti - u0996548

Related Work

The google movie-specific search results was our motivation to develop a more holistic approach to showing movies in a different perspective. Here are some of the google screenshots



Cast

View 10+ more


Tom Hanks
Chuck Noland


Helen Hunt
Kelly Frears

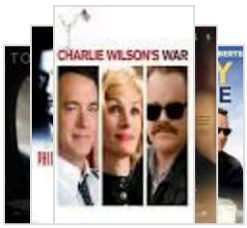

Lari White
Bettina Peterson


Nick Searcy
Stan


Chris Noth
Jerry Lovett

People also search for


Directed by Robert Zemeckis



Tom Hanks movies

Adventure movies

Questions

Through visualizing the data on movies, we are trying to answer the following questions:

- How many movies were made in a year?
- What movies did each actor play in?
- Which were the highest grossing movies of a year?
- What are the details about each movie like actors, director, genre, synopsis?
- How many movies had a particular range of ratings?
- How many movies had a particular range of social media awareness (facebook likes for movie pages, surprisingly old movies do have these) ?

By answering these questions, we will be able to get a big picture of trends in terms of evolution of movies over decades, how well movies have performed in their release year in terms of box-office results, current relevance (likes in social media etc.), duration, ratings etc.
The selective exploration of movies released every year also helps in diving more into the details and helping the user make informed opinions.

Data

- Our primary source of data was a curated dataset of over 5000 movies that we researched and found at kaggle.com
  - https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset

- Since people identify movies best with their posters, we wanted to include movie posters in our visualization, but we didn't want to store over 5000 images since that would take up serious amounts of space and hamper client loading performance. So we made use of a dynamic api provided by wemakesite creator Martin Ivanov. The api is available at
  - http://imdb.wemakesites.net/api/
  - http://omdbapi.com/
- The api returned json results like the following:
  -
```
{
    "status":"success",
    "code":200,
    "message":"ok",
    "data":{
        "id":"tt0115956",
        "type":"title",
        "title":null,
```

```
      "year":null,
      "description":"The pilot of a rescue copter, Captain Karen Walden, died shortly before her helicopter crew was rescued after it crashed in Desert Storm. It first  Written byBrian W Martz <B.Martz@Genie.com>",
      "certificate":"",
      "duration":"1h 56min",
      "released":"1996-07-12",
      "cast":[
         "Denzel Washington",
         "Meg Ryan",
         "Lou Diamond Phillips",
         "Michael Moriarty",
         "Matt Damon",
         "Bronson Pinchot",
         "Seth Gilliam",
         "Regina Taylor",
         "Zeljko Ivanek",
         "Scott Glenn",
         "Tim Guinee",
         "Tim Ransom",
         "Sean Astin",
         "Armand Darrius",
         "Mark Adair-Rios"
```

```
        ],
        "genres":[
           "action",
           "drama",
           "mystery",
           "thriller",
           "war"
        ],
        "directors":[

        ],
        "writers":[

        ],
        "image":"https:\/\/images-
na.ssl-images-
ama-
zon.com\/images\/M\/MV5BMTMwNDk0NjU
zOF5BMl5BanBnXkFtZTcwNjcxODIyMQ@@._
V1_UY268_CR2,0,182,268_AL_.jpg",
        "review":{
           "text":"Courage under fire
 was an okay movie, the story was g
ood enough to keep me : 6 out of 10
.",
           "rating":"7\/10"
        }}}
```

Data Processing

While our primary data was found from a reputable source, we still found some duplicate entries among the data source. In order to make it pluggable with any data set, we decided to implement our own logic to de-duplicate entries. This is done before any processing.

We were initially hitting the imdb api asynchronously whenever we wanted to load poster images, this involved a lot of network transfer. We decided we could pre-fetch the image url and tack that on to the data, so we did a run of data enhancement to enrich the data.

While enriching the data, we also found that there are other useful information from the response that was otherwise missing from our primary data source, so we merged our primary data with the newly found data.

Exploratory Data Analysis

We did our primary explorations of the data trying to find insights about movies reported in the dataset. We found that the dataset didn't have an intuitive way of looking at data. This led us to pursue additional data sources that are available online. We believe that people associate movies best with movie posters, so we downloaded a poster image which came up to 24KB in size. We had around 5000 movies in our data set, this would mean all the poster images for all the movies will come close to 1GB of data. **We didn't want to slow the client performance by loading 1GB of data**. Instead, we decided to hit an api to selectively load images that match the visualization filter and year filters we have in our visualization

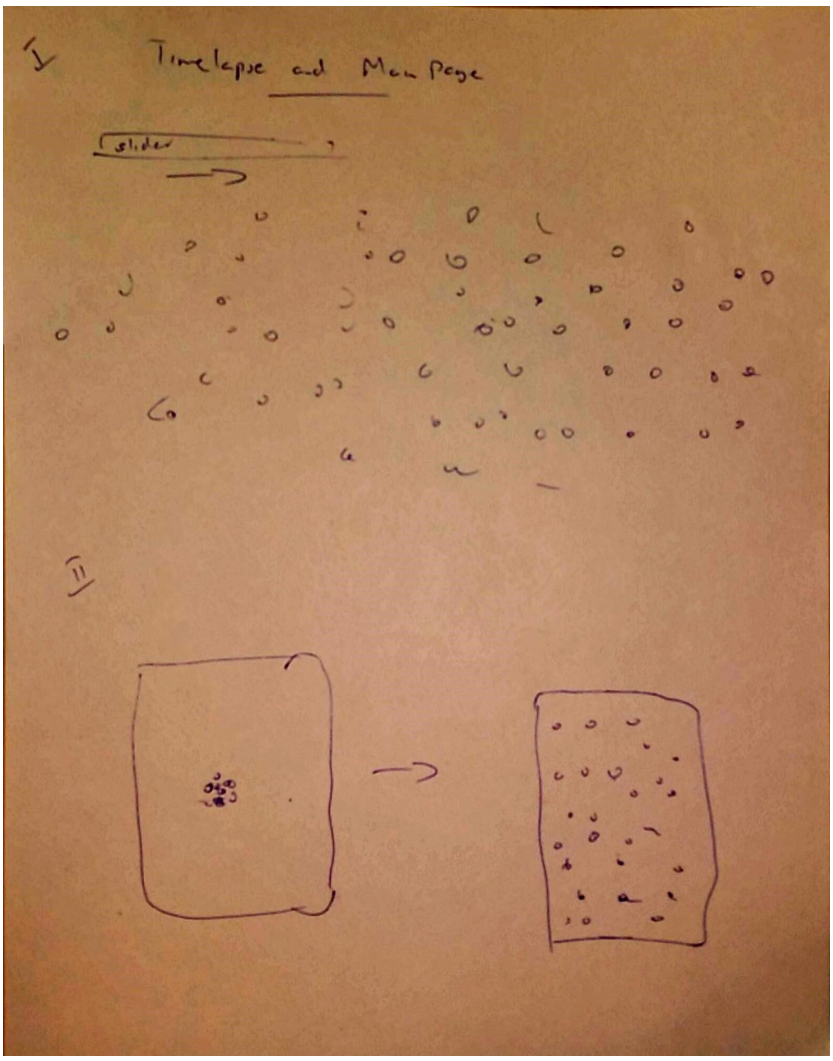
Enhancement of available data was proving to be another problem. We used two api end points to query the data to get additional details such as the poster images. First challenge we faced here was about **Synchronisation primitives in javascript.** We asynchronously hit data end points, but we needed additional scoreboarding to ensure safety and

linearizability of data. We ended up crashing the third party end points sending 1000s of requests at the same time, we had to use throttling to get the correct data.

Process Book – Boxoffice 360
Aniraj Kesavan – u0996550
Ashwini Janamatti - u0996548

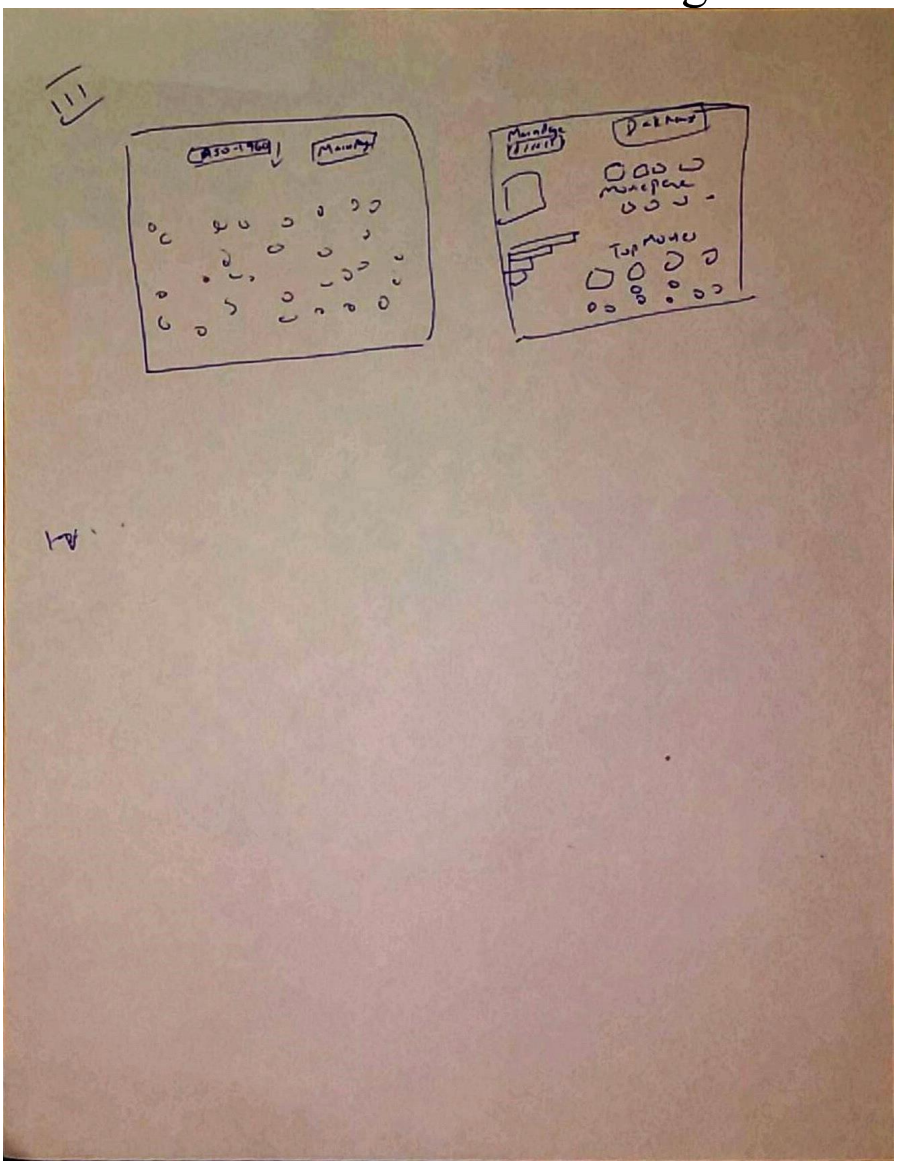
Design Evolution

Timelapse (Big Bang)



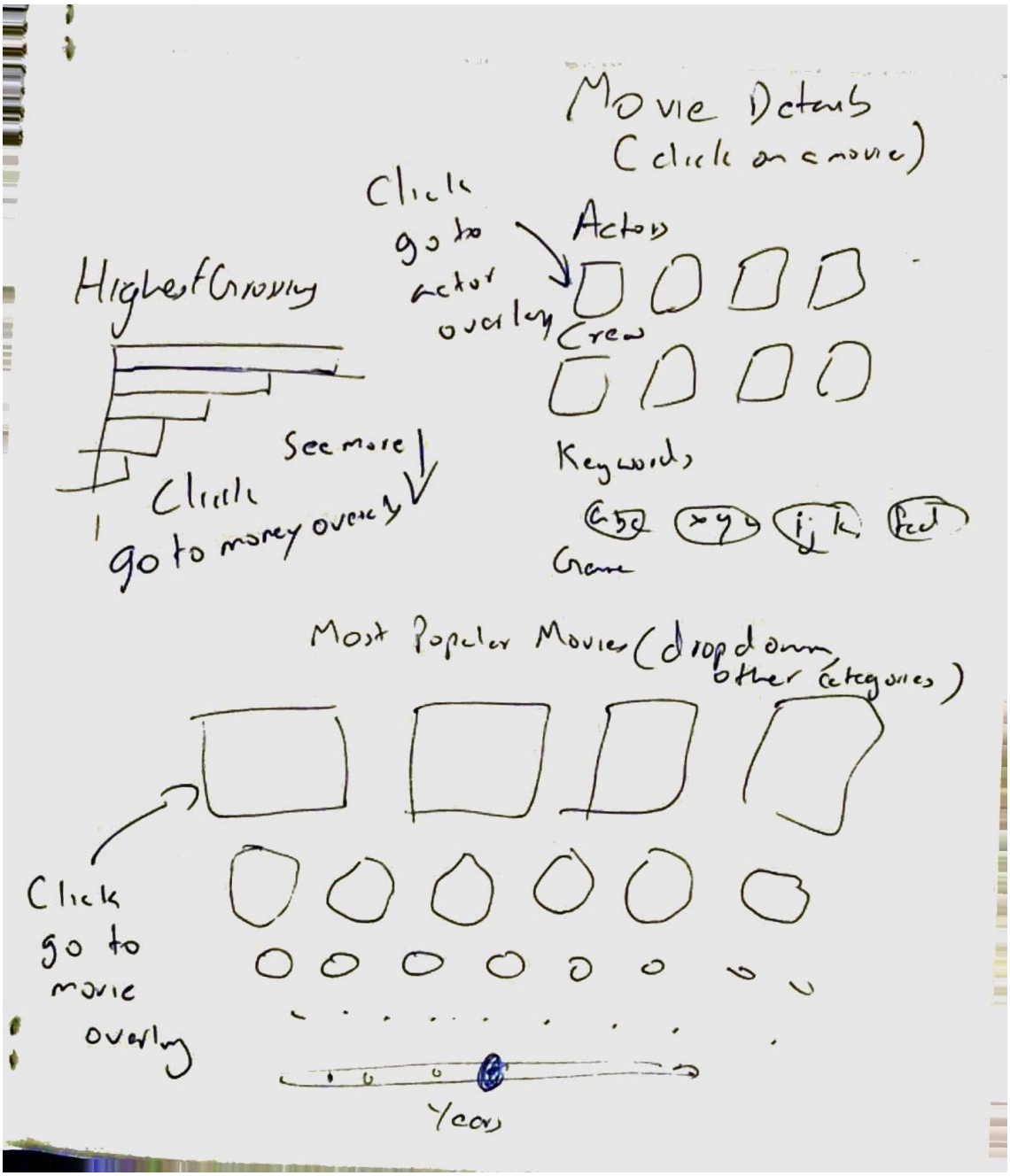
Our Initial thoughts around the big bang visualization. We wanted to show the evolution of movies overall. We wanted

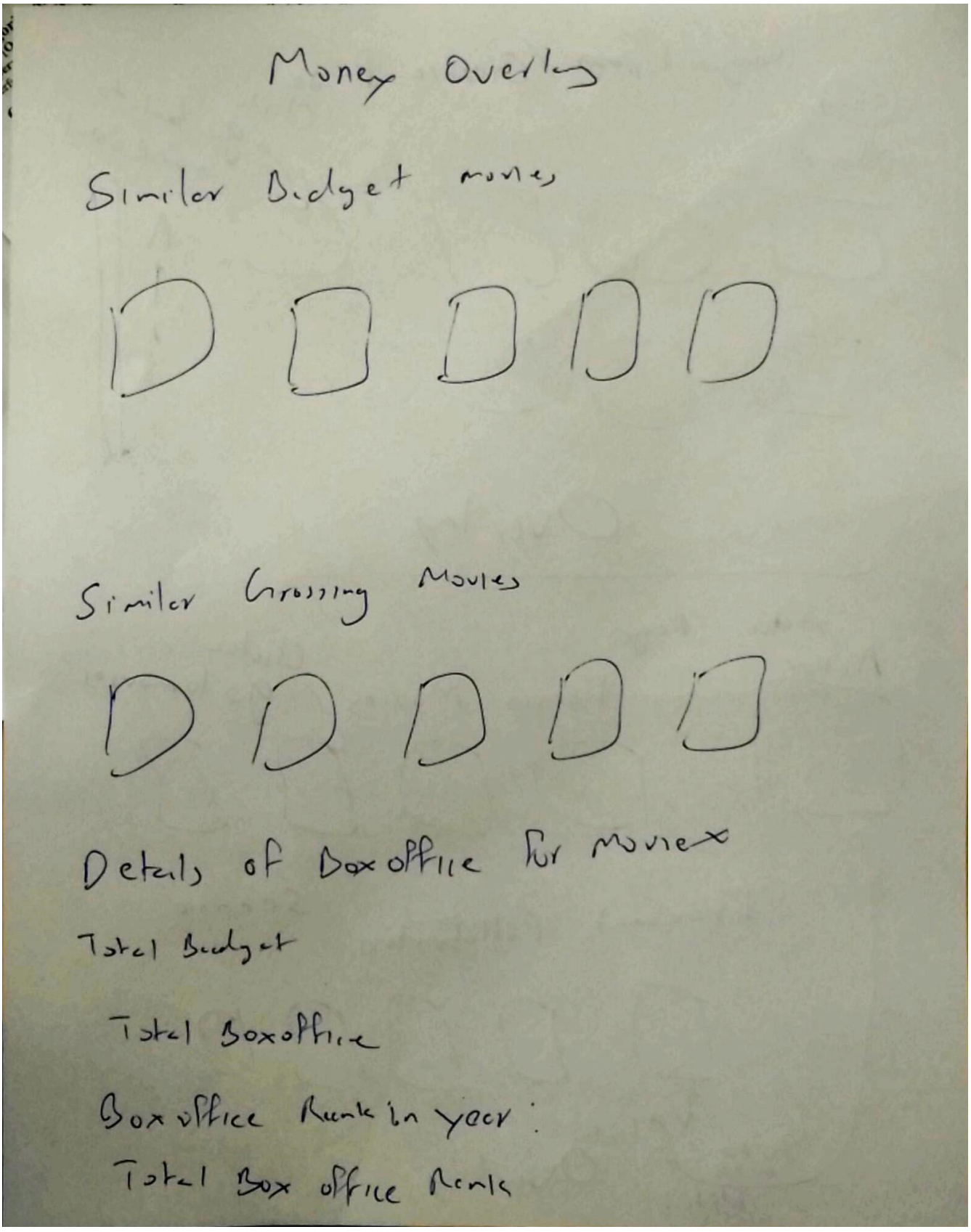
to start from a singularity and expand over to all 5000 movies in our dataset. While this was more aesthetically pleasing, it wouldn't offer meaningful insights to the user. Our next thoughts were to brush a period of movies and show a big bang for those years. Even then this was not giving enough information about the movies involved. We finally settled on showing per year visualization of the animation starting from a small cloud using d3's force animation and show the links between actors and movies using this.

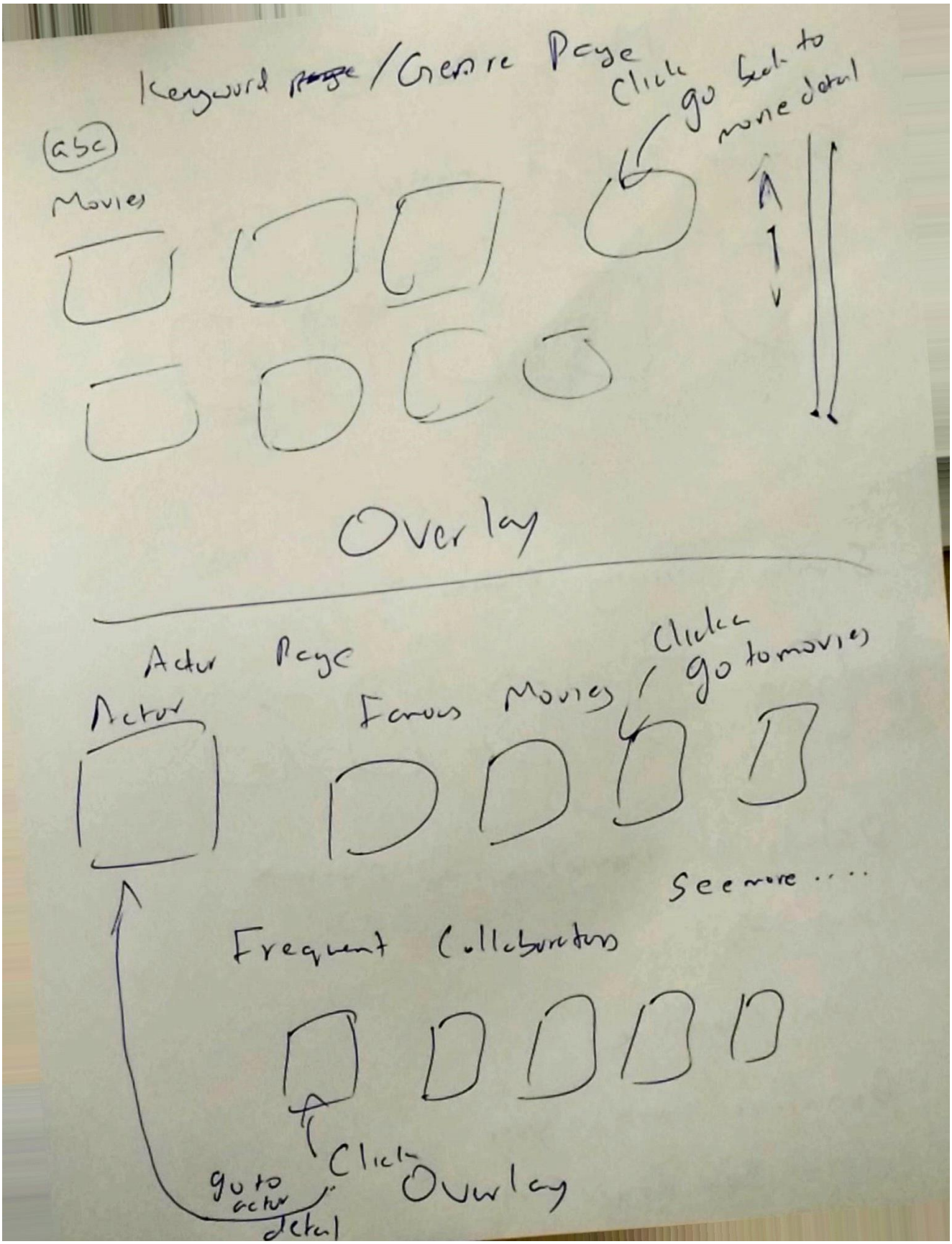


Per year Visualization

We wanted to enable the user to do detailed data exploration using per year movie visualizations. We decided to have a main tab which shows most of the details .
We were ambitious to have a navigational context showing overlay tabs. We didn't
end up implementing those due to time constraints. Our initial thoughts were that
a user would be able to pick up cues from actors which goes to personell tabs showing actors and their famous movies. Picking up keywords would show an overlay tab showing all the movies in the genre/having those keywords. There was also supposed to be a "money" tab which navigates from the bar chart.


Movie Details
(click on a movie)
Click
go to
actor
overlay
Actors
Crew
Highest Grossing
See more
Click
go to money overlay
Keywords
abc
xyz
ijk
feel
Genre
Most Popular Movies (drop down other categories)
Click
go to
movie
overlay
Years


Money Overlay
Similar Budget movies
Similar Grossing Movies
Details of Boxoffice for movie
Total Budget
Total Boxoffice
Boxoffice Rank in year:
Total Box office Rank


Keyword / Genre Page
Click go back to movie detail
(asc)
Movies
Overlay
Actor Page
Actor
Famous Movies
Click go to movies
See more ....
Frequent Collaborators
Click Overlay
go to actor detail

Must-Have features

We decided to separate out must have features and optional features. The key visualizations we wanted to show were:

1. Time-lapse (Big Bang)
   The big bang animation needed to be shown in order to visualize the general trends of evolution of movies over the years. This was key in understanding the growth of number of movies and the industry
2. Per year Visualisation
   Since we couldn't show more details in a holistic manner, we decided to add granularity of years as a separate visual tab. This was really important to show the plethora of information associated with the movies.
   We arrived at the conclusion that too many visual elements will be confusing, so we decided on showing top 10 movies of any selected criteria as "tiles". The tiles needed to be sorted according to various criteria such as:
   - Imdb Rating
   - Movie facebook likes
   - Duration
   - Budget

3. Detail pane. Once we show the tiles conforming to selected sort order and filters, we also needed a detail pane which shows all the details pertaining to a single movie. On clicking a movie tile, the detail pane should reflect changes to the selection made.

Optional features

1. Show Actor-Movie relation in Big Bang
   The big bang animation could link between actors and movies. If we could filter only movies where a particular actor was involved in a given year, we could also show the link between movies and actors.

2. A bar chart showing the highest grossing movies
   For a given year, it would be nice to have a bar chart which shows the top 5 highest grossing movies in every year.

3. Rearranging tiles (per year visualization) and nodes (big bang)
   Since our visualizations change in response to a year slider, it will make sense if there could be a pleasant animation showcasing the change.

4. Overlay tabs: On selecting a movie, if the detail pane could offer navigation to tabs such as the one for actors or genre, it would be conversational in nature.

Implementation

We implemented our visualization using javascript, html and css. We made use of jquery and d3 library and borrowed parts of implementation of a d3 slider from an open source implementation. We asynchronously hit third party apis to extract additional information and to load dynamic poster images. We implemented the two key visualizations (Big Bang and Per Year) and presented them in a tabbed manner. We did our preprocessing of data entirely in JavaScript.

1. Time-lapse

The key technique we used in order to generate the big bang was d3's force simulation. We generate a timeline processing the csv movie dataset, on detecting a change on the slider, we bombard the canvas with actors and movie nodes and link them based on their relationship (If actor a acted in movie m, there will be a link between a and m nodes). Then we do a big bang on the cluster of objects using force simulation. We also provide the functionality to filter nodes based on actors. We provide a drop down list of actors, which if selected will show movies that the particular actor was associated with in a

given year.

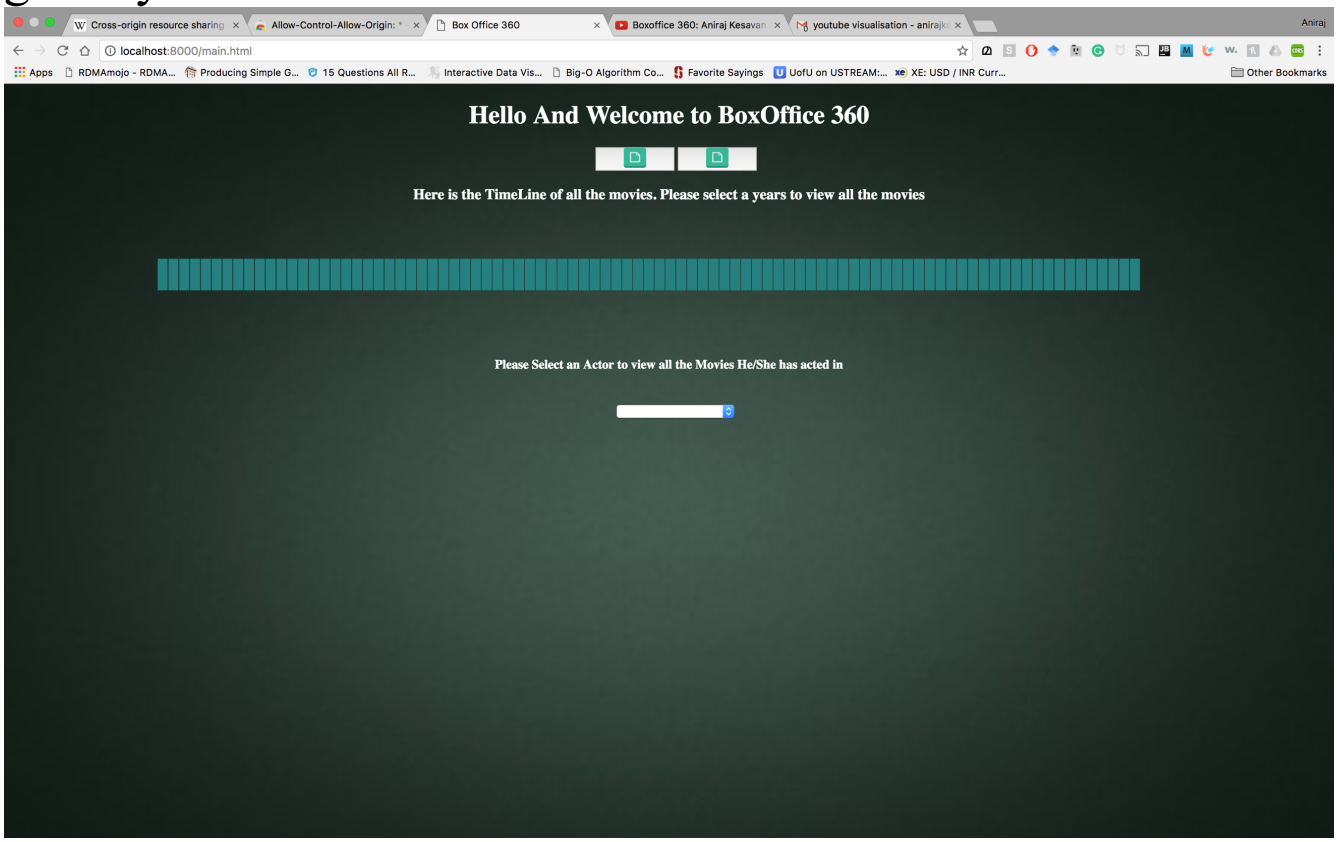


Fig1: Welcome page

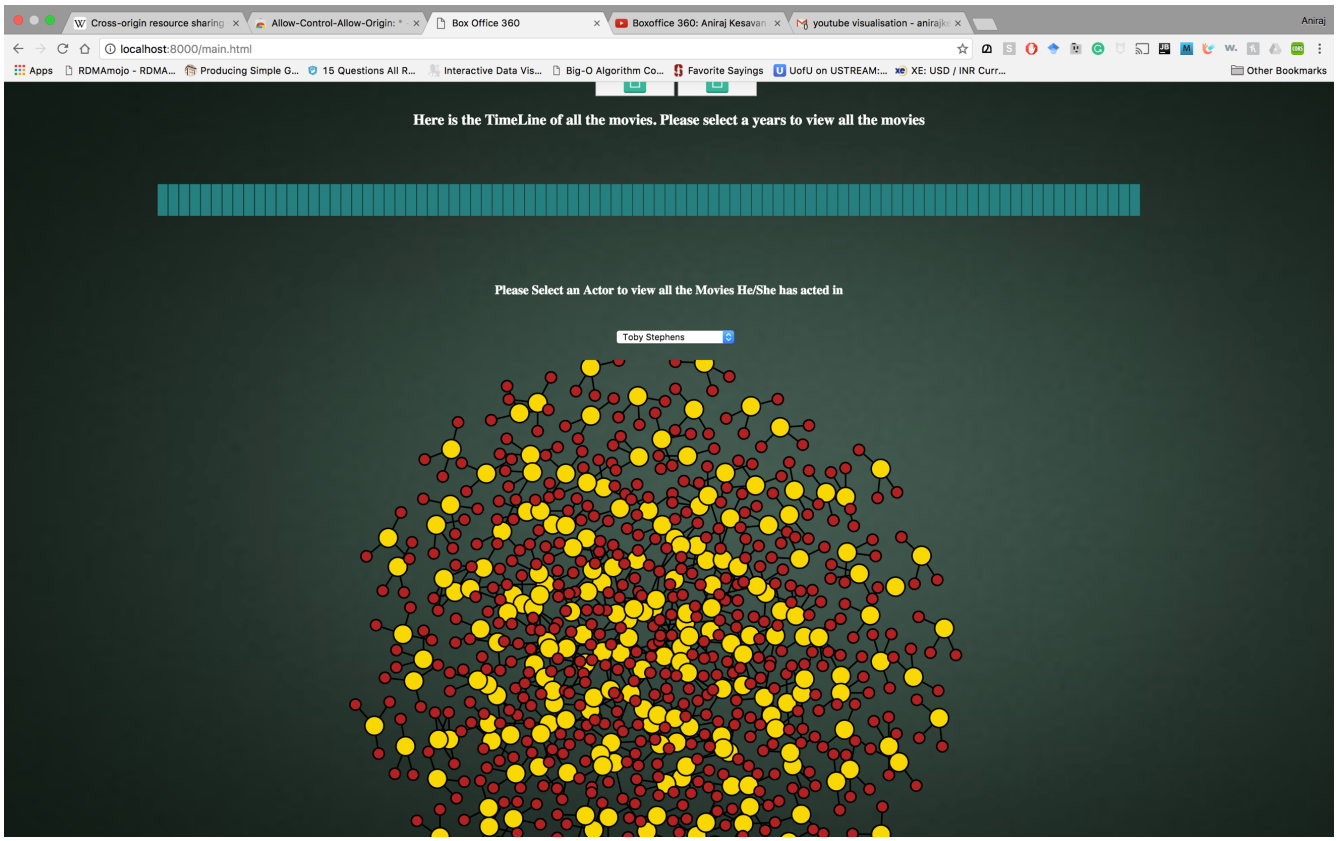


Fig2: Showing Big bang of movies(yellow nodes) and actors (red nodes)

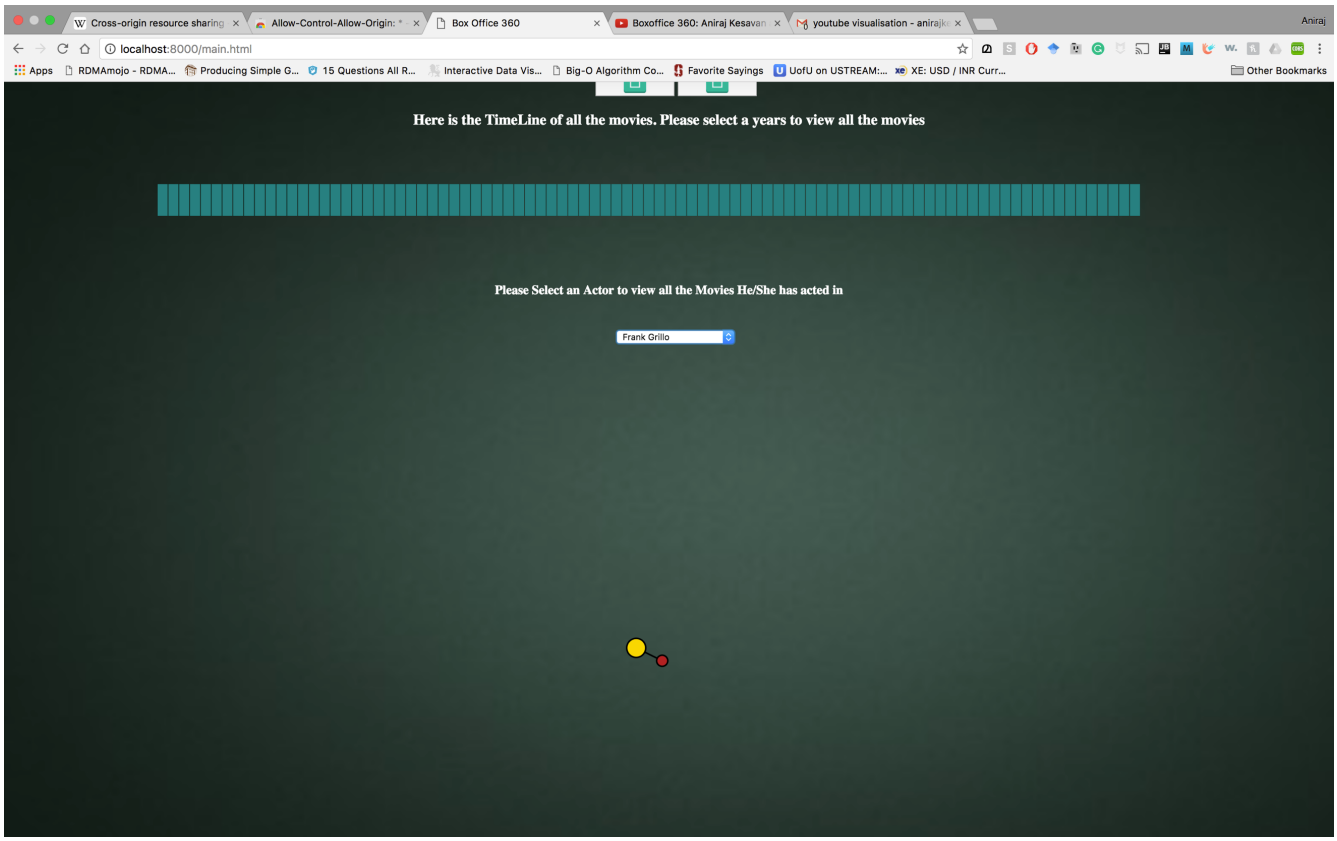


Fig3: Filtered based on actor name

2. Per Year Visualization.

The key technique behind per year visualization were to asynchronously load poster images based on the selected sort and filter criteria. We used ajax queries for asynchronous loading of images and used vanilla JavaScript for the basic implementation. We made use of d3 selectors to select elements in the visualization. The visualization starts from a d3 slider implementation of year slider. Once you select an year,

we populate first 10 movies in our dataset in the movie tiles asynchronously by contacting third party end points to get images using the movie title's unique id (Imdb resource id). We grab additional attributes of data during preprocessing that enables us to provide an option to sort movies based on these attributes. The movie tiles are replaced with newer tiles that fit the selected criteria. We also provide range based filters to filter movies by their imdb rating and social media perception (total likes garnered by the movie's facebook page). There is also a bar chart that gets populated once you select the year. The bar chart shows the top 5 highest grossing movies for a selected year. For further exploration of data, we can click on a movie tile which will populate a detail pane with details such as the key crew members, synopsis of the movie, the genre and other significant details. This is also dynamically updated when we change the selection by clicking on a movie tile.

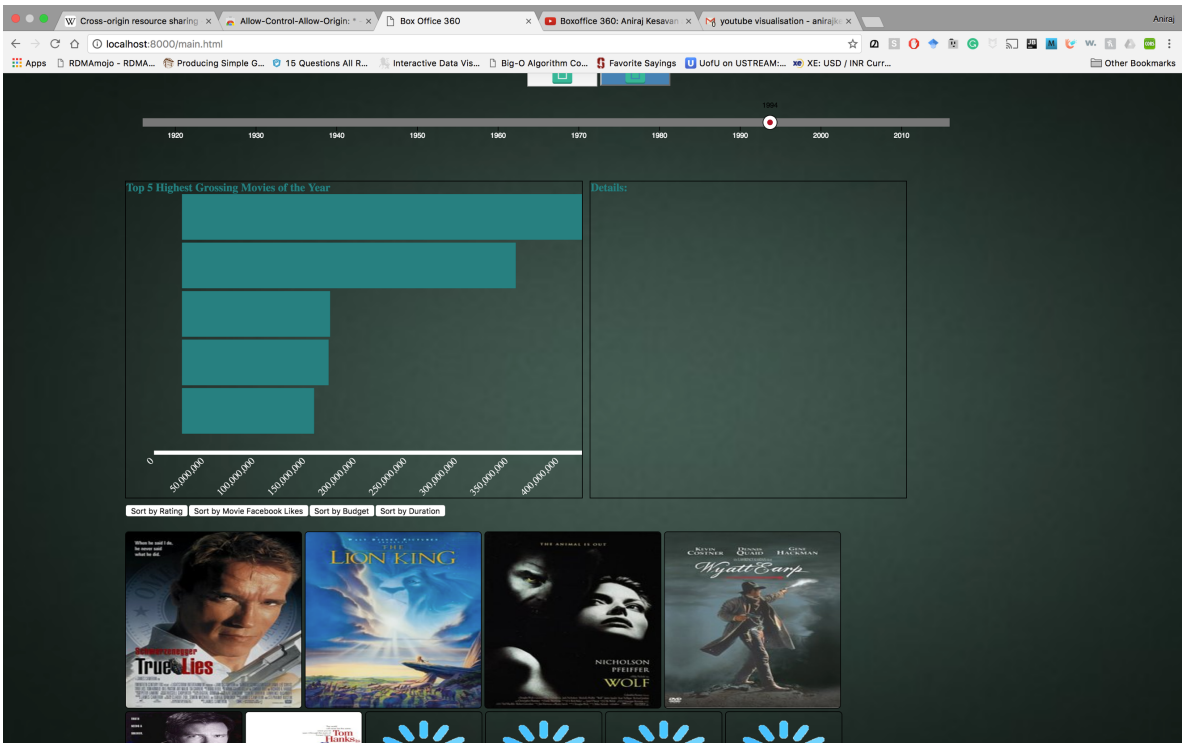


Fig4: Selected year 1994

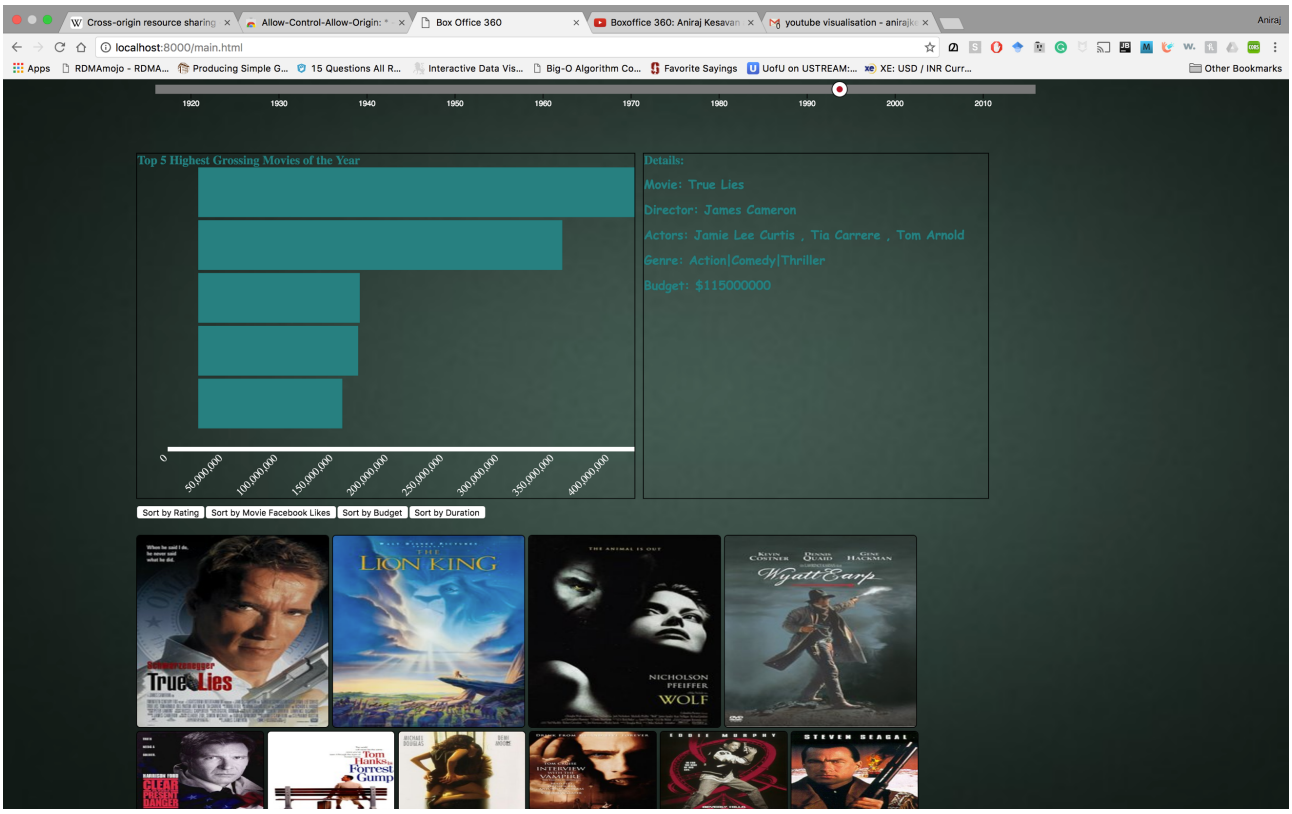


Fig5: Detail pane populated based on clicked movie tile

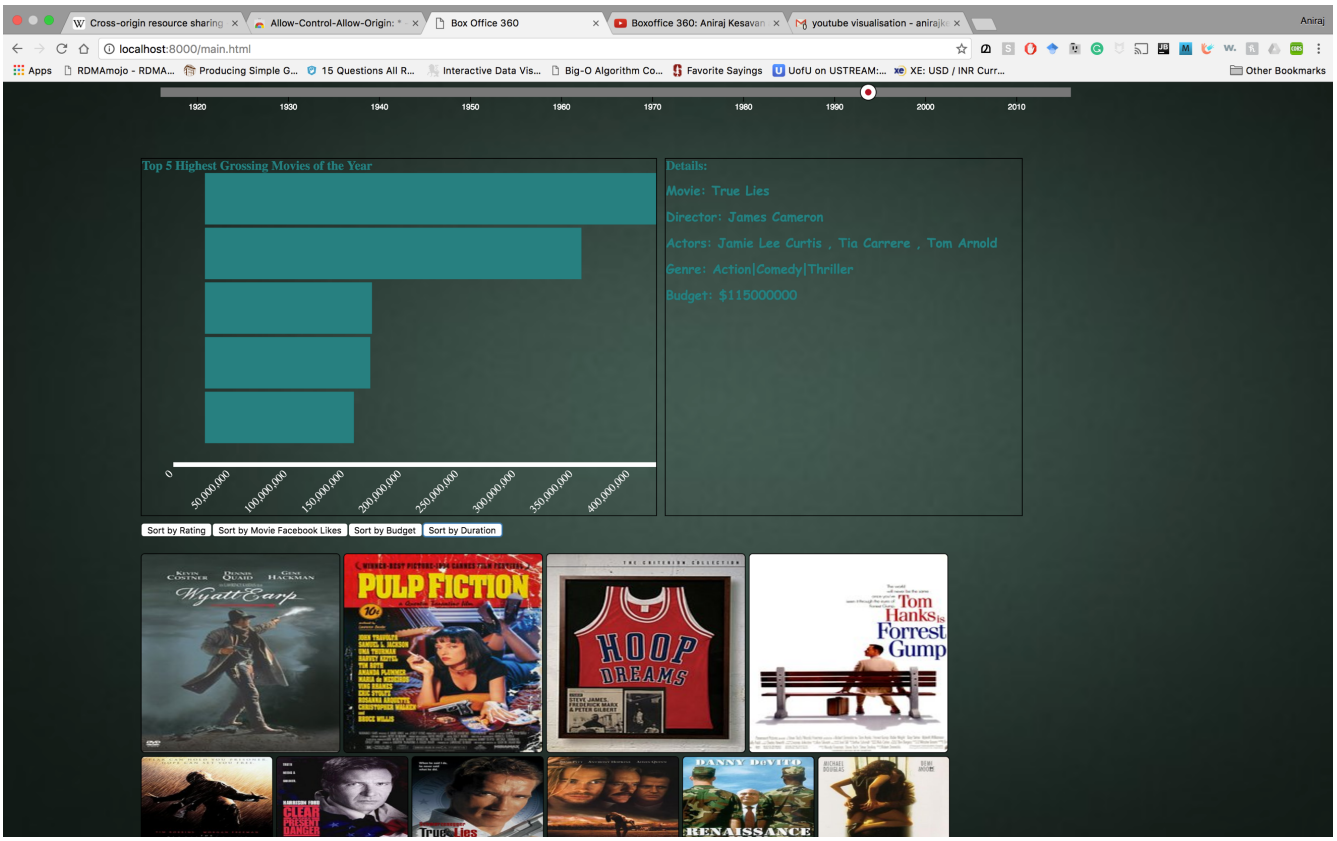


Fig6: Tiles sorted by duration

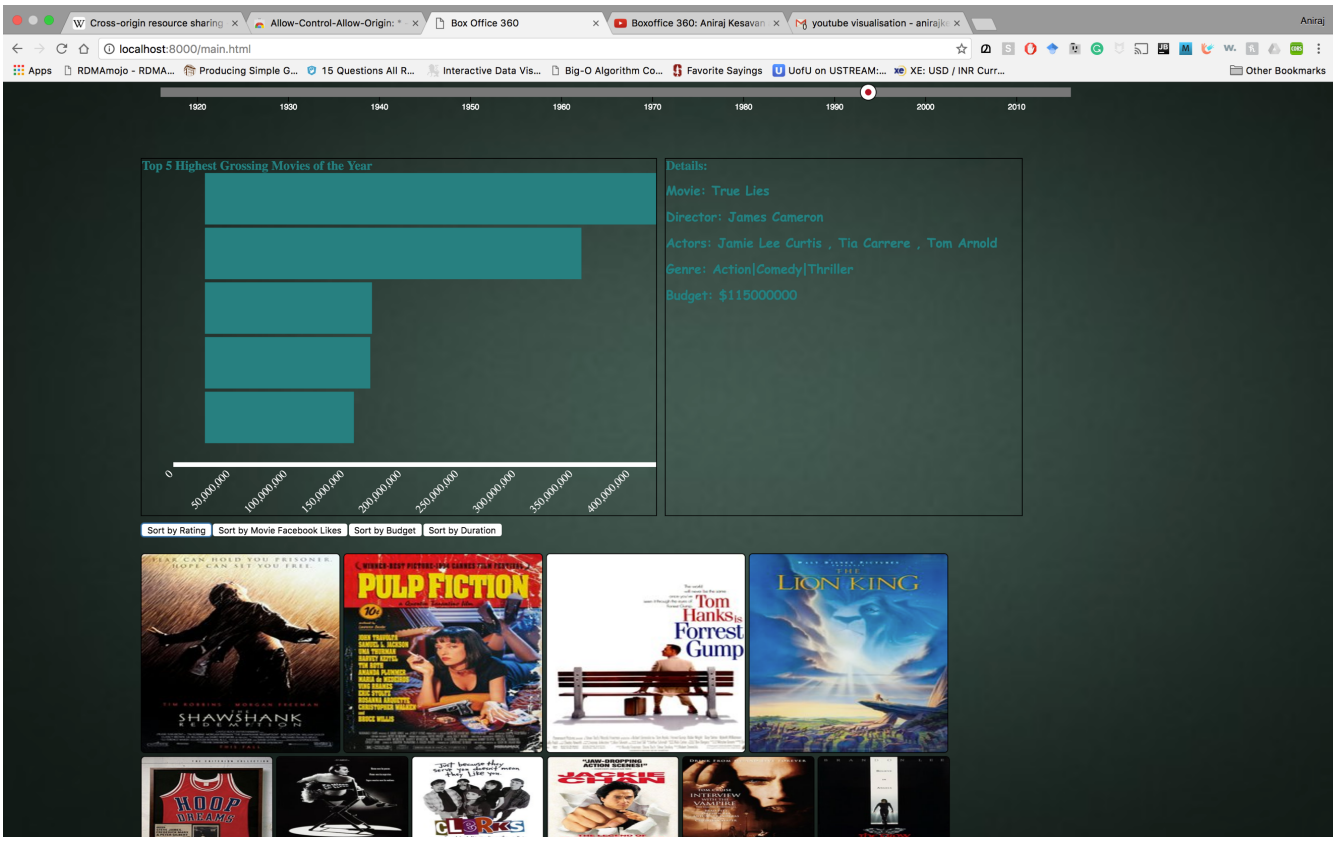


Fig7: Tiles sorted by imdb rating.

Evaluation

General Trends

The general trends we found from the exploration of the data are the following:

As you move towards the recent years, number of movies (that were recorded in imdb) goes up. In the early years, there were gaps in the data points from our data sets, but if you visualize the big bang closely, you can see that the space gets really crowded moving to the 2000s.

Popularity of the movie and it's critical rating are closely correlated. If you filter based on higher ends of popularity and critic's rating, you get similar results in your tiles.

Social media relevance of the movies doesn't

correlate to their release timeline. We have found recent movies which either doesn't have a facebook page or there exists a page with less number of likes than old classics. People relive and reconnect with old movies which were box-office successes as well as the ones with raving critical reviews.

Future Work

We found that while our dataset explores the general trend over the years, it is by no means exhaustive. We did join our data with two additional sources using third party APIs, but we could generate more interesting visualizations if we could find more data sources.

One other optional feature that we had in mind which we couldn't get to implement was the overlay tabs for navigation.