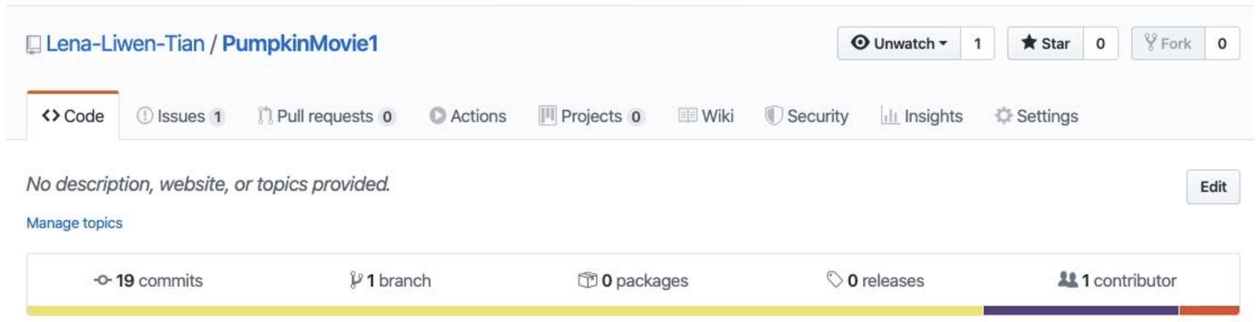
Technical Report for Phase One

Website link

https://master.d120nc9yxmrxsg.amplifyapp.com/

Github link

https://github.com/Lena-Liwen-Tian/PumpkinMovie1



No description, website, or topics provided.

19 commits · 1 branch · 0 packages · 0 releases · 1 contributor

Team Information

Name:Liwen Tian
Email:liwentian1103@gmail.com
Github username: Lena-Liwen-Tian
Canvas group: group 8
Project name: Pumpkin Movie

Project Description

My project name is Pumpkin Movie, it provides users a better way to buy their ideal movie tickets online. On my website, users can find movie information in details, eg. movie time, movie genres, movie ratings. Users can also get the relevant information about all theatres in the United States, eg.location, phone numbers. Besides, the show time of different movies at different theatres are also listed on the website for users to choose from. Pumpkin Movie allows users to combine information about movies, theatres and show time ahead of time, so that they can buy an ideal movie ticket in the end. The target users are young people who are looking for an easier way to buy an ideal ticket in order to get a better viewing experience in the theatre.

User Stories

In Phase 2, my user stories are shown as follows.

1. As a user, I want to use the calendar to select the movie I want to watch, so that I can plan my time.

I estimated that I would spend about a day doing it since it requires too much knowledge I am not familiar with. Actually, I was so lucky that I found a Sample online showing how to fill dates into cells and then render them. I spent about 8 hours on it. The assumption for this user story is users need to plan their time before buying the movie ticket.

2. As a user, I want to see the map of different theatres so that I can know how to go there.

I estimated that I would spend about 5 hours on it, but I spent 1.5 day on it in the end. I failed to render the Modal(a framework for the window) a lot of times at the beginninig, then I looked through BootStrap and found the answer. The assumption for this is that users are used to online GoogleMap to look for the place.

3. As a user I want to be able to see which theaters are showing the movie I am interested in when I look at the Movie Page, so that I can make a better choice.

I estimated that I would spend about 5 hour on it and I did spend about 5 hours. The assumption for this is that the users prefer to make decisions after collecting relevant information.

4. As a user, I want to see the price of the movie on my online ticket so that I can know how much I need to spend on it.

I estimated that I would spend about 1 hour on this, but actually I spent 40 minutes on this. This is my last step in this Phase and It is not hard for me. The assumption is that users prefer to check the price before making the purchase.

5. As a user, I want to know the recent showtimes in different theatres on Theatre Page so that I can select the time I want easily.

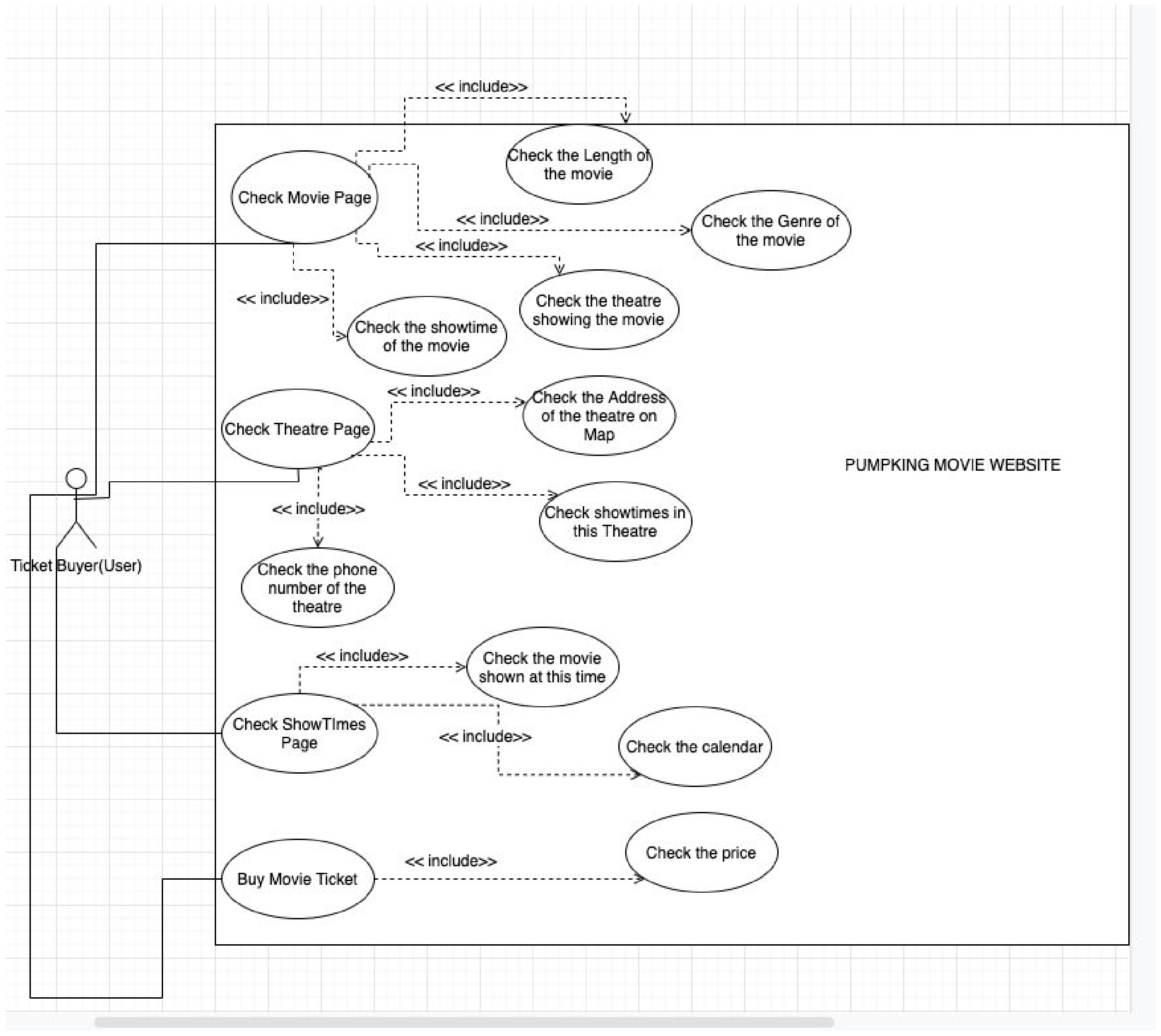
I estimated that I would spend about 30 minutes on this. I spent about 20 minutes. I did this together with user story 3 and they follow the same framework. The assumption is that users care about both the location and showtimes when buying the ticket.

6.As a user, I want to see at least three pages of movies for me to choose so that I can find the one I like. (Extra Feature)

I estimated that I would spend about 5 hours on it. Actually,I spent 0.5 day. The assumption is that users always want to have more choices.

Use Case Diagram

My use case diagram is shown below. Based on the user story above, the User(can also be called TicketBuyer) is an actor, he/she has associations with the Pumpkin Movie Website. He/she has about 5 new use cases in this Phase. I combined them with the use cases in Phase 1. I added “buy movie tickets” since I included the button of buying a movie ticket in my website though I have not made them dynamically yet.

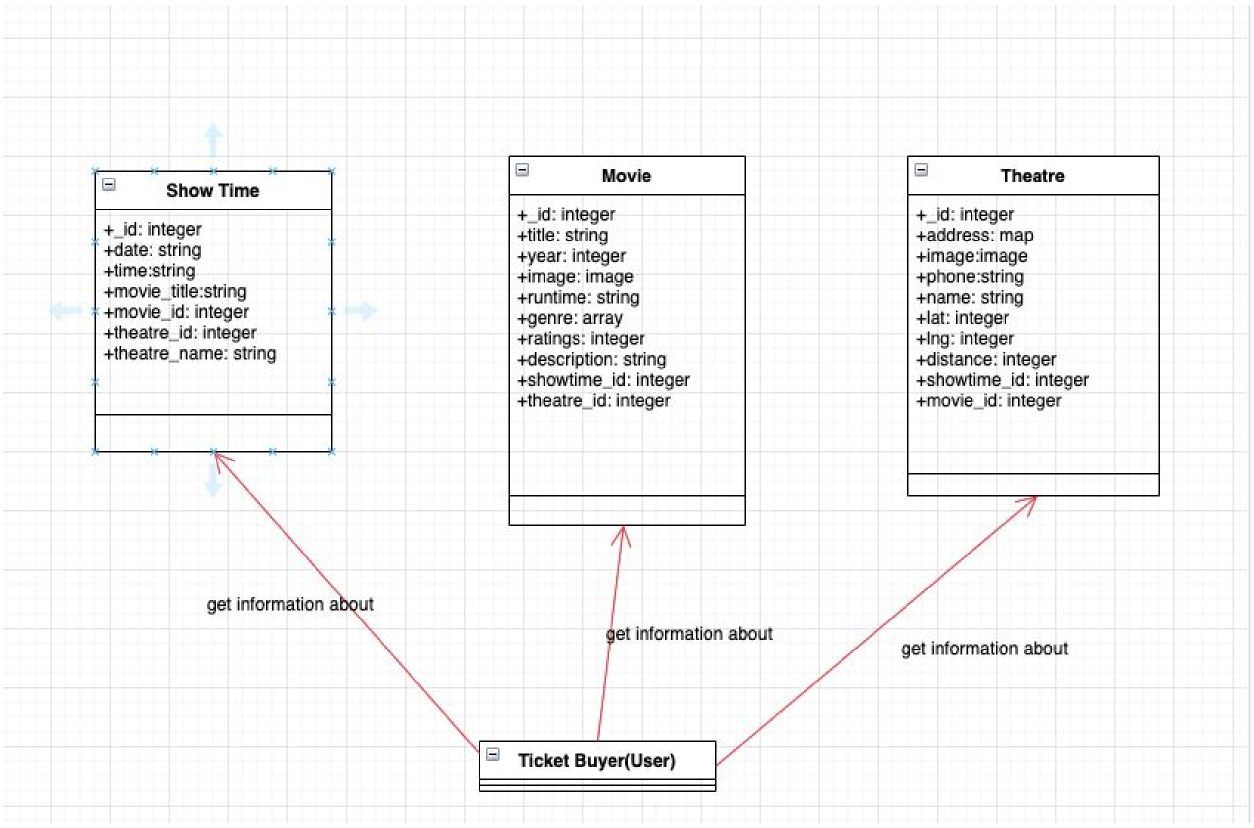


PUMPKING MOVIE WEBSITE
Ticket Buyer(User)
Check Movie Page
Check the Length of the movie
Check the Genre of the movie
Check the theatre showing the movie
Check the showtime of the movie
Check Theatre Page
Check the Address of the theatre on Map
Check showtimes in this Theatre
Check the phone number of the theatre
Check ShowTImes Page
Check the movie shown at this time
Check the calendar
Buy Movie Ticket
Check the price
<< include>>

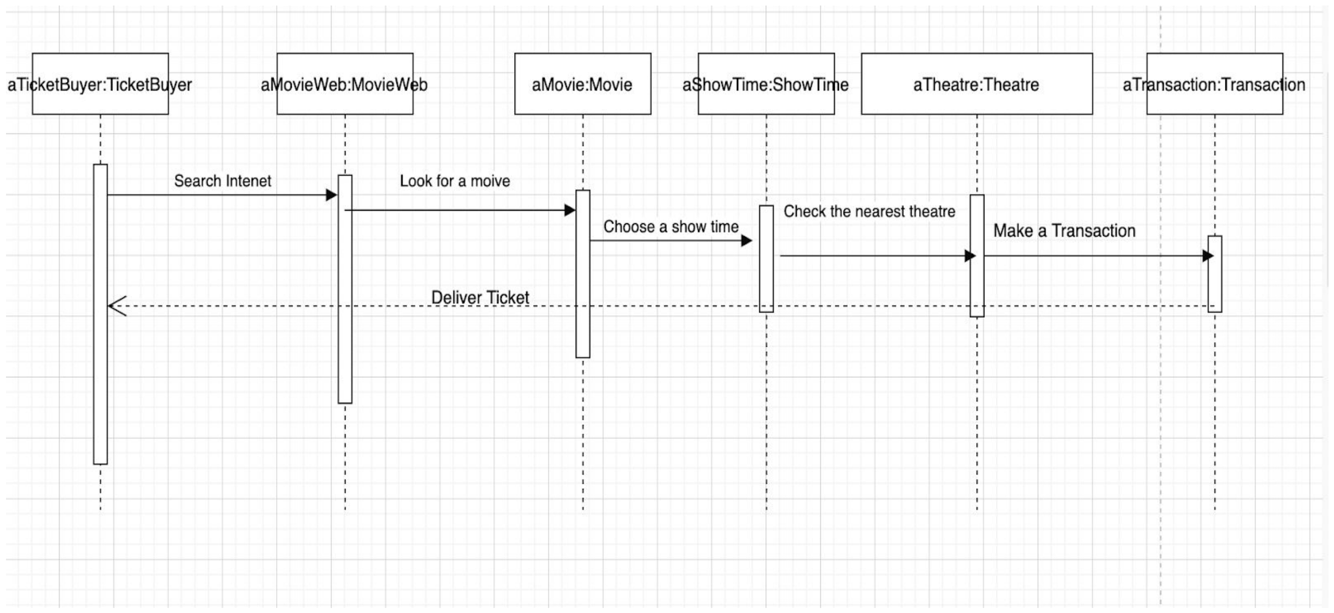
Design and UML Diagrams

The general idea of Pumpkin Movie is to let users get the information about movies, theatres and showtimes ahead of time, then they can buy their ideal tickets in the end. The first UML diagram shows three models I created. I added some features of different models in this Phase.

The Movies model has 10 fields: id, title, year, image, runtime, genre, ratings, description, showtime_id and theatre_id. Genre is an array because one movie may have two genres. The Show Time model has 7 fields: id, date, time, movie_title, theatre_name, movie_id and theatre_id. Movie and Theatre are related to each other through the Show Time. The Theatre model has 10 fields: id, address, image, phone, name, latitude, longitude, distance, showtime_id and movie_id. Now three models are related to each other. The Ticket Buyer(User) shown at the bottom can get the information from all three models above.

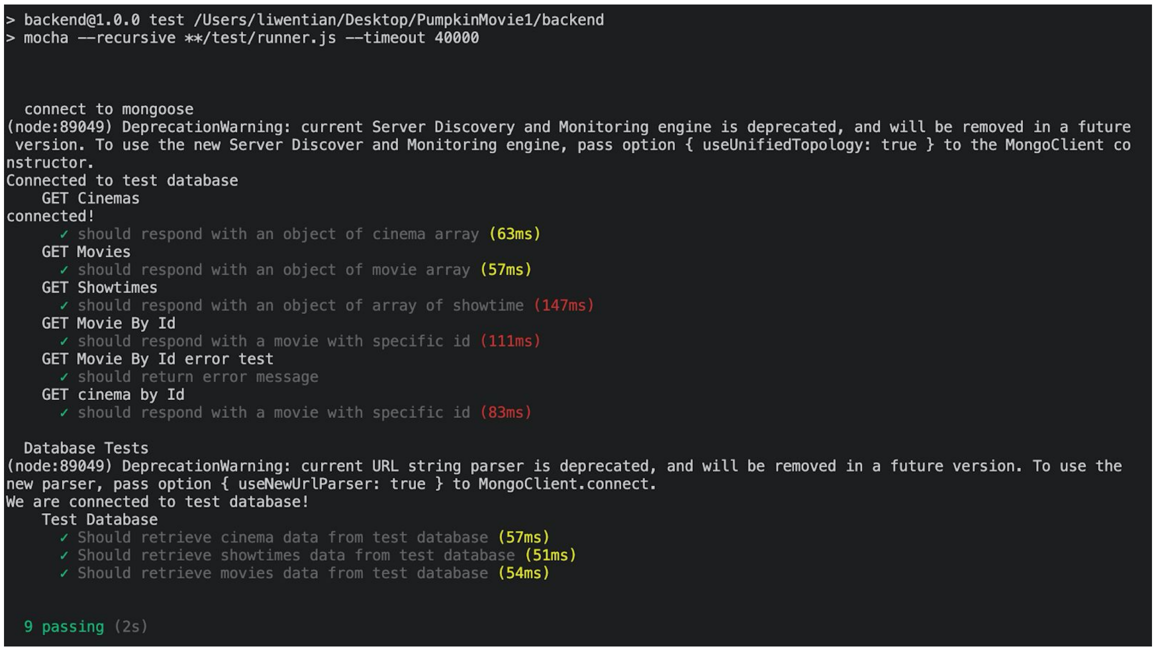


The ideal sequential actions for the user visiting this website in my mind is still similar to what I thought about in Phase1. Though I did not add any restrictions among the model connections now, I still want users to get the information step by step instead of getting lost in too much information at hand. As I mentioned before, In order to buy an ideal movie ticket, the first step for users is to search the Internet and open our website. Then they can click the movies button and look through the movies page to see which movie they want to watch. After picking up the movie they want, they click “buy a ticket” at the bottom of the movie page. This button can direct them to the show time page which shows users the time and locations for a certain movie. Suppose the user is not familiar with the location of different theatres, they need to check the location of the different theatres after picking up the right time. After all things are done, the user has chosen the time, location and movie for their ticket, they will be redirected to the transaction page, complete the ticket purchase online. In the end, the ticket will be delivered to the user online. Below is the Sequential Diagram. I have not realized it in this phase, but it shows the transaction process I will complete later.



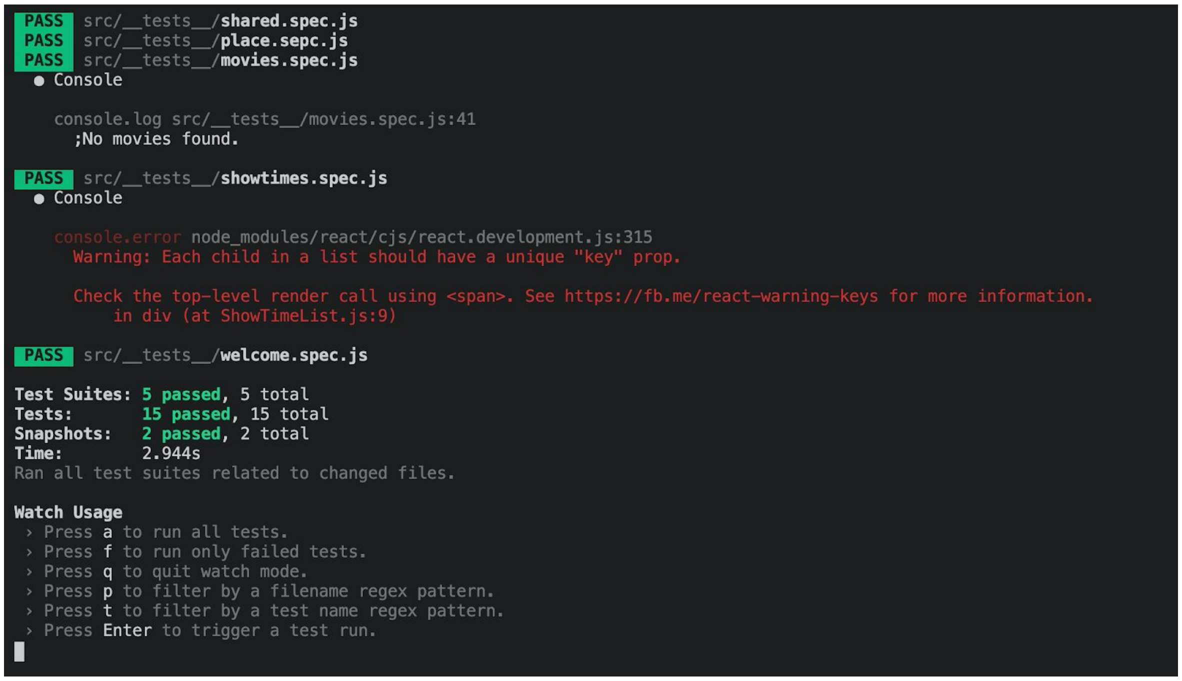
Testing

This is my first time to test the project. I spent too much time on it. I used Mocha and Chai to test the Nodejs codes I wrote for the backend. I tested all routes to see if I can take different actions on different routes. I also tested the models I created. I used the fake database in Mongodb to see if I can connect to it and got the right information I want through the models I wrote in the backend. I did 9 tests for it in total and all of them were successful in the end.



```
> backend@1.0.0 test /Users/liwentian/Desktop/PumpkinMovie1/backend
> mocha --recursive **/test/runner.js --timeout 40000



  connect to mongoose
(node:89049) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future
 version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient co
nstructor.
Connected to test database
    GET Cinemas
connected!
      ✓ should respond with an object of cinema array (63ms)
    GET Movies
      ✓ should respond with an object of movie array (57ms)
    GET Showtimes
      ✓ should respond with an object of array of showtime (147ms)
    GET Movie By Id
      ✓ should respond with a movie with specific id (111ms)
    GET Movie By Id error test
      ✓ should return error message
    GET cinema by Id
      ✓ should respond with a movie with specific id (83ms)

  Database Tests
(node:89049) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version. To use the
new parser, pass option { useNewUrlParser: true } to MongoClient.connect.
We are connected to test database!
    Test Database
      ✓ Should retrieve cinema data from test database (57ms)
      ✓ Should retrieve showtimes data from test database (51ms)
      ✓ Should retrieve movies data from test database (54ms)


  9 passing (2s)
```

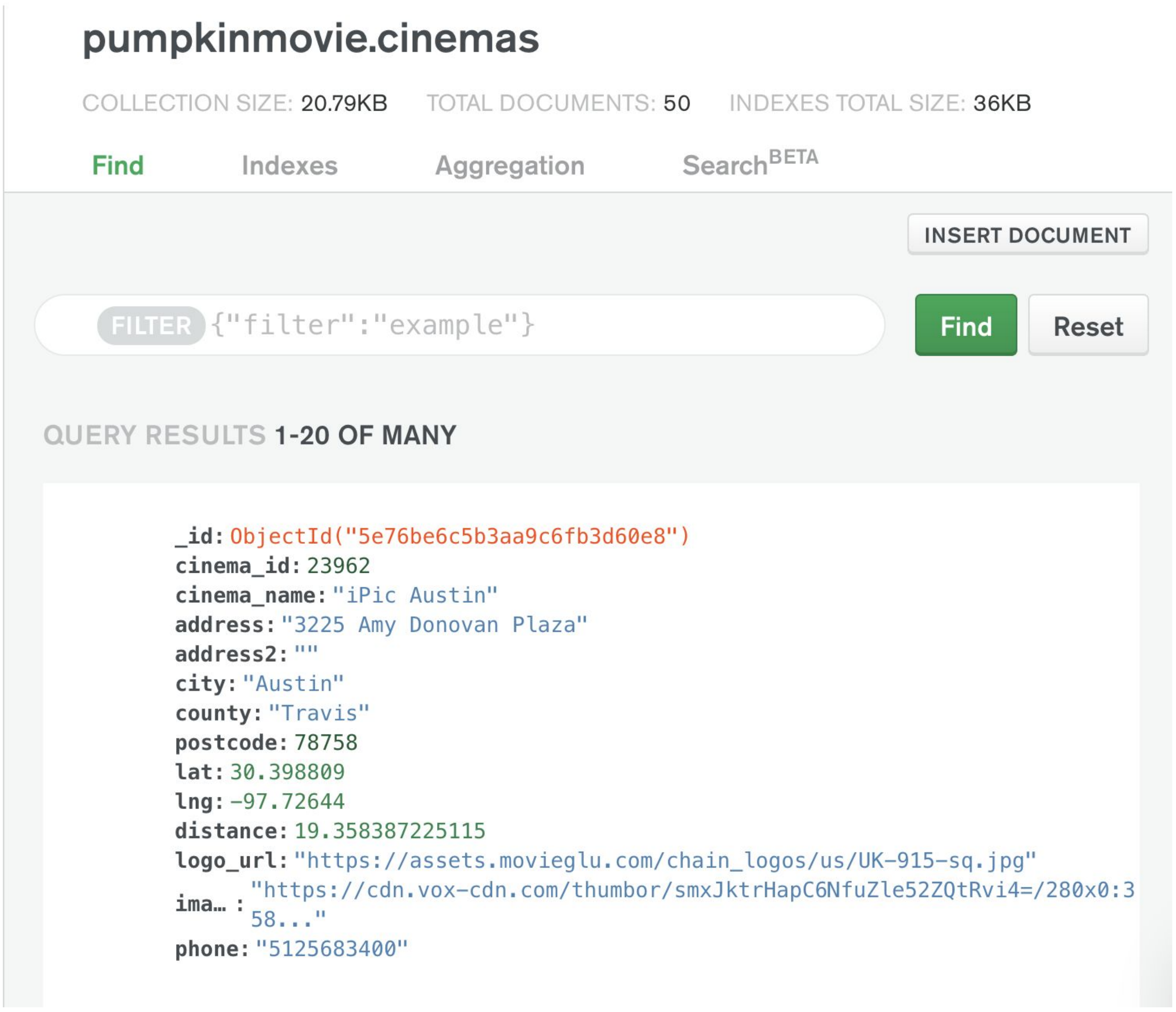
I planned to only use Mocha to test the frontend at the very beginning, but I found that it is not enough for front end testing. I wrote frontend in react. There are too many components and interactions. I incorporated Jest and Enzyme since they are good at mocking fetch and parameters. Testing methods change a lot these years. I found Shallow and Mount are popular in recent times. I mostly used these two methods to test different components. I completed about 15 in total.



```
 PASS  src/__tests__/shared.spec.js
 PASS  src/__tests__/place.sepc.js
 PASS  src/__tests__/movies.spec.js
  ● Console

    console.log src/__tests__/movies.spec.js:41
      ;No movies found.

 PASS  src/__tests__/showtimes.spec.js
  ● Console

    console.error node_modules/react/cjs/react.development.js:315
      Warning: Each child in a list should have a unique "key" prop.

      Check the top-level render call using <span>. See https://fb.me/react-warning-keys for more information.
          in div (at ShowTimeList.js:9)

 PASS  src/__tests__/welcome.spec.js

Test Suites: 5 passed, 5 total
Tests:       15 passed, 15 total
Snapshots:   2 passed, 2 total
Time:        2.944s
Ran all test suites related to changed files.

Watch Usage
 › Press a to run all tests.
 › Press f to run only failed tests.
 › Press q to quit watch mode.
 › Press p to filter by a filename regex pattern.
 › Press t to filter by a test name regex pattern.
 › Press Enter to trigger a test run.
▌
```

Models

I have three models from three sources. I have already fetched all information I wanted from the different databases and related them. My screenshots of different models in Mongodb are shown below.
This is the Cinema Model, it has id, cinema_id, cinema_name, address, address2, city, county, postcode, latitude, longitude,distance, logo-url, image, phone. I collected 50 documents for it.

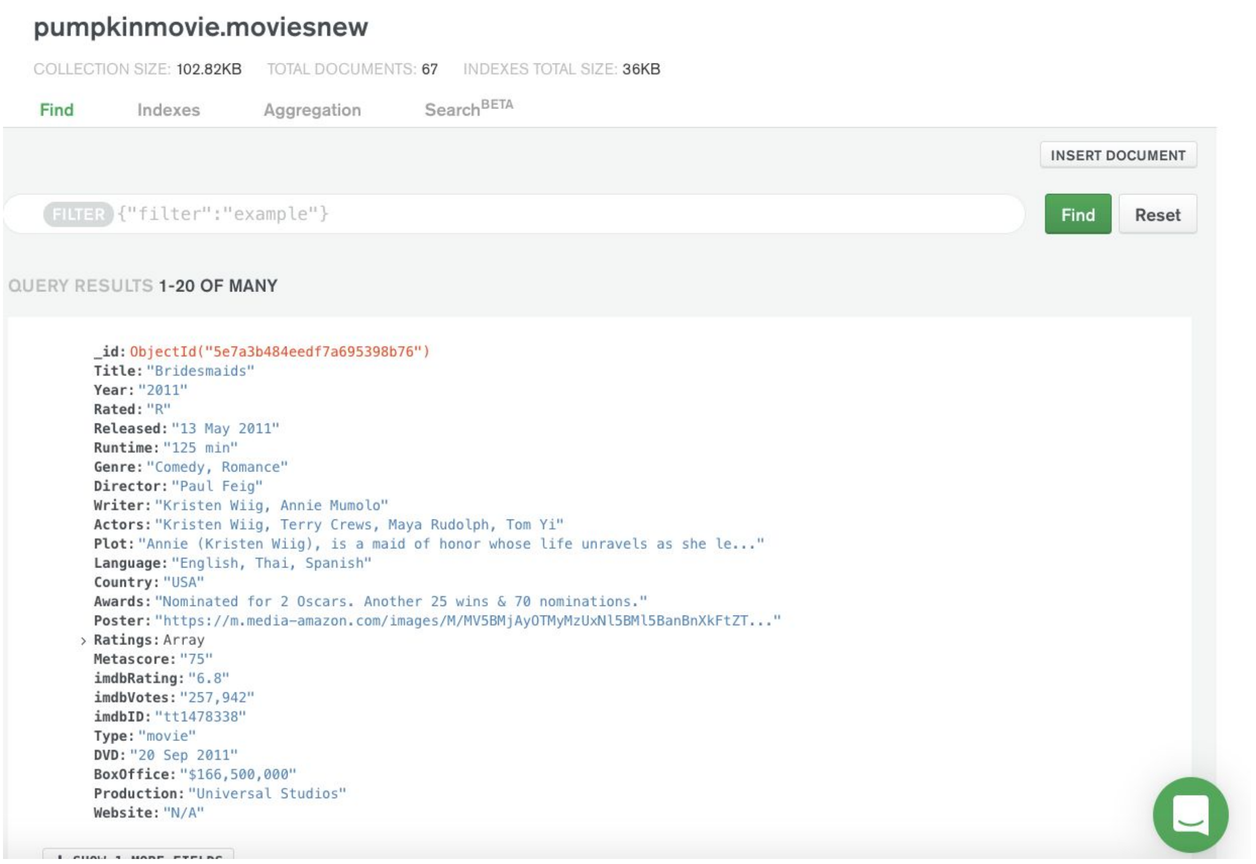


```
_id: ObjectId("5e76be6c5b3aa9c6fb3d60e8")
cinema_id: 23962
cinema_name: "iPic Austin"
address: "3225 Amy Donovan Plaza"
address2: ""
city: "Austin"
county: "Travis"
postcode: 78758
lat: 30.398809
lng: -97.72644
distance: 19.358387225115
logo_url: "https://assets.movieglu.com/chain_logos/us/UK-915-sq.jpg"
ima… : "https://cdn.vox-cdn.com/thumbor/smxJktrHapC6NfuZle52ZQtRvi4=/280x0:3
       58..."
phone: "5125683400"
```

I got these data from MovieGlueDatabase. I used Python to Fetch data through their API.

```
  {
    "cinema_id": 8893,
    "cinema_name": "Cineworld Leicester Square",
    "address": "Leicester Square",
    "city": "London",
    "county": "Greater London",
    "postcode": "WC2H 7NA",
    "lat": 51.51078,
    "lng": -0.1305,
    "distance": 0.031255415281335,
    "logo_url": "https://d2z9fe5yu2p0av.cloudfront.net/chainlogos/uk/UK-868-sq.jpg"
  },
  {
    "cinema_id": 8923,
    "cinema_name": "Vue Cinemas - West End",
    "address": "3 Cranbourn Street",
    "city": "London",
    "county": "Greater London",
    "postcode": "WC2H 7AL",
    "lat": 51.511181,
    "lng": -0.12934,
    "distance": 0.064288623403495,
    "logo_url": "https://d2z9fe5yu2p0av.cloudfront.net/chainlogos/uk/UK-712-sq.jpg"
```

The second Model is Movie. I collected about 70 documents for it and cleaned it to 67. It has id, Title, Year, Rated, Released, Runtime, Genre, Director, Writer, Actors, Plot, Language, Country, Awards, Poster, Ratings, metascore, imdbRating, imdbVotes, imdbID, Type, DVD, BoxOffice, Production, Website. I got these from IMDB.



pumpkinmovie.moviesnew

COLLECTION SIZE: 102.82KB TOTAL DOCUMENTS: 67 INDEXES TOTAL SIZE: 36KB

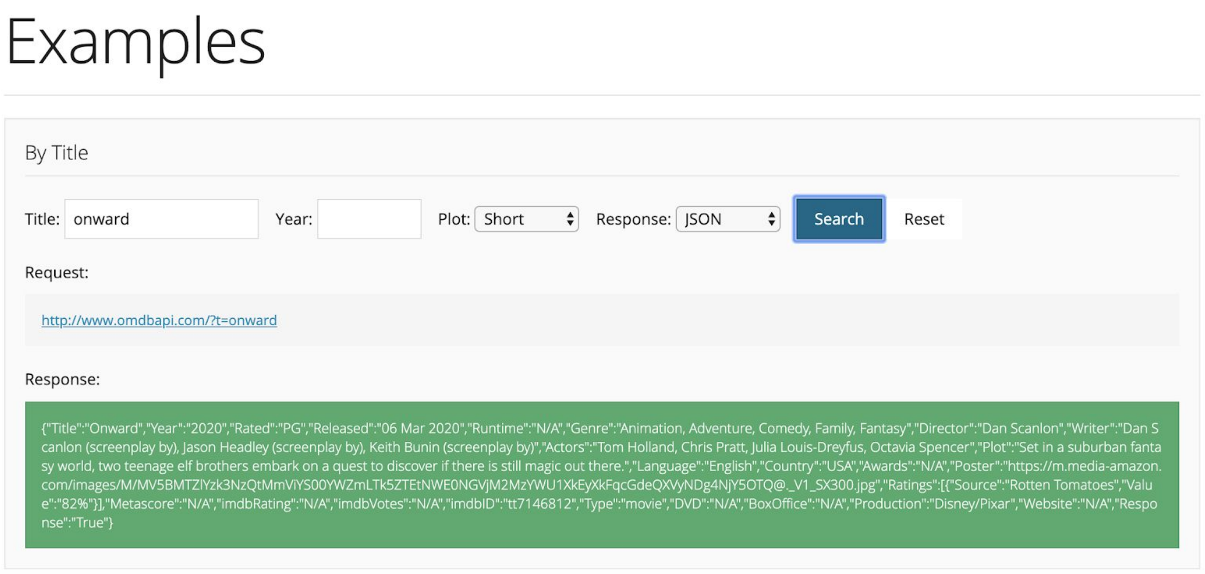
Find Indexes Aggregation Search BETA

INSERT DOCUMENT

FILTER {"filter":"example"} Find Reset

QUERY RESULTS 1-20 OF MANY

```
_id: ObjectId("5e7a3b484eedf7a695398b76")
Title: "Bridesmaids"
Year: "2011"
Rated: "R"
Released: "13 May 2011"
Runtime: "125 min"
Genre: "Comedy, Romance"
Director: "Paul Feig"
Writer: "Kristen Wiig, Annie Mumolo"
Actors: "Kristen Wiig, Terry Crews, Maya Rudolph, Tom Yi"
Plot: "Annie (Kristen Wiig), is a maid of honor whose life unravels as she le..."
Language: "English, Thai, Spanish"
Country: "USA"
Awards: "Nominated for 2 Oscars. Another 25 wins & 70 nominations."
Poster: "https://m.media-amazon.com/images/M/MV5BMjAyOTMyMzUxNl5BMl5BanBnXkFtZT..."
> Ratings: Array
Metascore: "75"
imdbRating: "6.8"
imdbVotes: "257,942"
imdbID: "tt1478338"
Type: "movie"
DVD: "20 Sep 2011"
BoxOffice: "$166,500,000"
Production: "Universal Studios"
Website: "N/A"
```



Examples

By Title

Title: onward Year: Plot: Short Response: JSON Search Reset

Request:

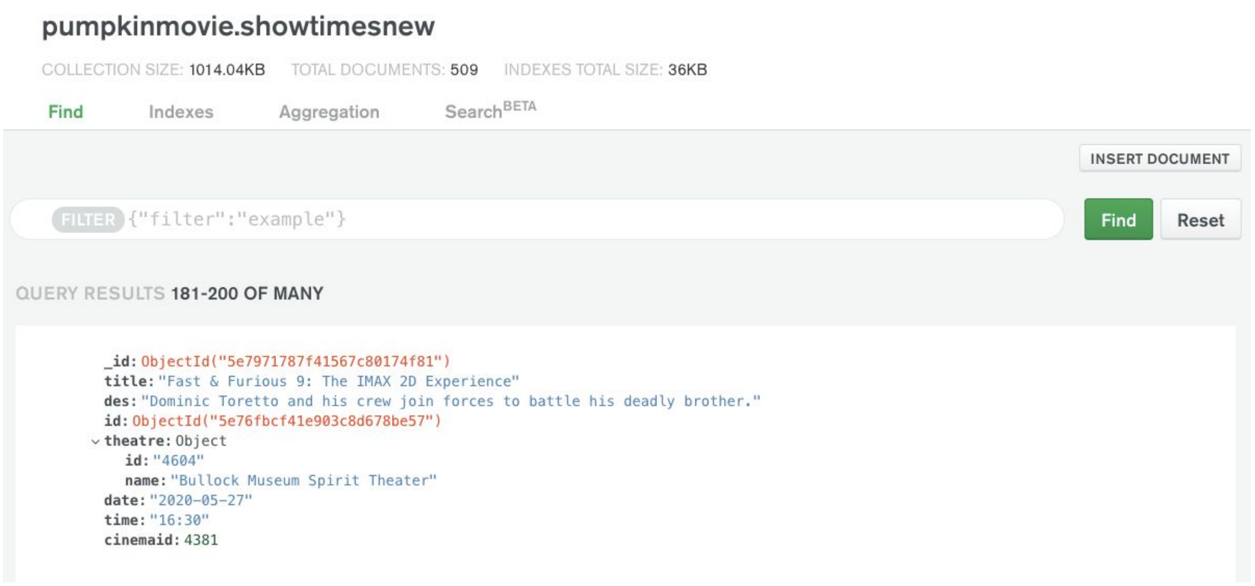
http://www.omdbapi.com/?t=onward

Response:

{"Title":"Onward","Year":"2020","Rated":"PG","Released":"06 Mar 2020","Runtime":"N/A","Genre":"Animation, Adventure, Comedy, Family, Fantasy","Director":"Dan Scanlon","Writer":"Dan Scanlon (screenplay by), Jason Headley (screenplay by), Keith Bunin (screenplay by)","Actors":"Tom Holland, Chris Pratt, Julia Louis-Dreyfus, Octavia Spencer","Plot":"Set in a suburban fantasy world, two teenage elf brothers embark on a quest to discover if there is still magic out there.","Language":"English","Country":"USA","Awards":"N/A","Poster":"https://m.media-amazon.com/images/M/MV5BMTZlYzk3NzQtMmViYS00YWZmLTk5ZTEtNWE0NGVjM2MzYWU1XkEyXkFqcGdeQXVyNDg4NjY5OTQ@._V1_SX300.jpg","Ratings":[{"Source":"Rotten Tomatoes","Value":"82%"}],"Metascore":"N/A","imdbRating":"N/A","imdbVotes":"N/A","imdbID":"tt7146812","Type":"movie","DVD":"N/A","BoxOffice":"N/A","Production":"Disney/Pixar","Website":"N/A","Response":"True"}

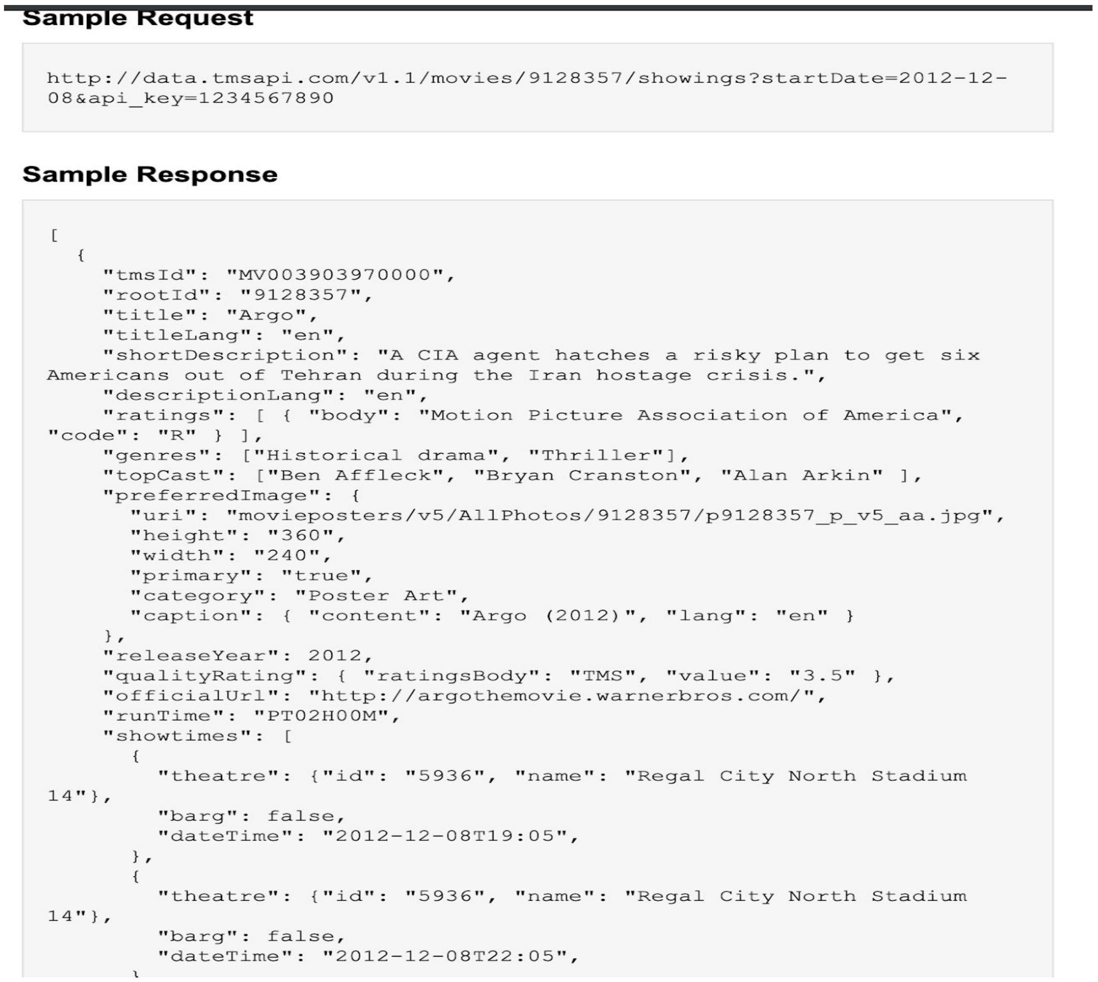
The third Model is Show Time. Movies and theatres are related to each other through showtimes. I got showtimes in local theatres(Texas) in the recent 90 days. However, a lot of theatres closed due to Cronovirus. I did not get as much data as I expected. I collected about

200 documents for it. I got these data from GraceNote Developer I used their sample php codes to fetch the data.


pumpkinmovie.showtimesnew
COLLECTION SIZE: 1014.04KB TOTAL DOCUMENTS: 509 INDEXES TOTAL SIZE: 36KB
Find Indexes Aggregation SearchBETA
INSERT DOCUMENT
FILTER
Find Reset
QUERY RESULTS 181-200 OF MANY

```
{"filter":"example"}
```

```
_id: ObjectId("5e7971787f41567c80174f81")
title: "Fast & Furious 9: The IMAX 2D Experience"
des: "Dominic Toretto and his crew join forces to battle his deadly brother."
id: ObjectId("5e76fbcf41e903c8d678be57")
theatre: Object
  id: "4604"
  name: "Bullock Museum Spirit Theater"
date: "2020-05-27"
time: "16:30"
cinemaid: 4381
```



Sample Request

```
http://data.tmsapi.com/v1.1/movies/9128357/showings?startDate=2012-12-
08&api_key=1234567890
```

Sample Response

```
[
  {
    "tmsId": "MV003903970000",
    "rootId": "9128357",
    "title": "Argo",
    "titleLang": "en",
    "shortDescription": "A CIA agent hatches a risky plan to get six
Americans out of Tehran during the Iran hostage crisis.",
    "descriptionLang": "en",
    "ratings": [ { "body": "Motion Picture Association of America",
"code": "R" } ],
    "genres": ["Historical drama", "Thriller"],
    "topCast": ["Ben Affleck", "Bryan Cranston", "Alan Arkin" ],
    "preferredImage": {
      "uri": "movieposters/v5/AllPhotos/9128357/p9128357_p_v5_aa.jpg",
      "height": "360",
      "width": "240",
      "primary": "true",
      "category": "Poster Art",
      "caption": { "content": "Argo (2012)", "lang": "en" }
    },
    "releaseYear": 2012,
    "qualityRating": { "ratingsBody": "TMS", "value": "3.5" },
    "officialUrl": "http://argothemovie.warnerbros.com/",
    "runTime": "PT02H00M",
    "showtimes": [
      {
        "theatre": {"id": "5936", "name": "Regal City North Stadium
14"},
        "barg": false,
        "dateTime": "2012-12-08T19:05",
      },
      {
        "theatre": {"id": "5936", "name": "Regal City North Stadium
14"},
        "barg": false,
        "dateTime": "2012-12-08T22:05",
```

I copied some of my Python Scripts for cleaning and organizing the database.

Tools

I use Reactjs to build the frontend and Nodejs for the backend. I also included BootStrap for some React Components like Modal, and Validation. To fetch data from the websites, I used Python and php. I used Python to clean and organize my database. The testing tools in this Phase are Mocha, Jest, Chai, Enzyme and Selenium. Mocha for front end testing, Selenium for GUI and all others for backend testing. The software I used in this phase is Visual Studios. This is my first time using it to create a Project. It is not very hard. I used AWS Amplify together with Heroku to deploy my website. I tried other AWS services a lot of times but all of them failed in the end.

Reflections

I am becoming more and more familiar with MERN now. I think this Project is almost done. I still want to add some extra features like making transactions online and writing online reviews for different movies. I am trying to figure them out now. What I feel struggled about is testing. It is too tedious and complicated. I haved used Mocha before. I left some days for testing but still get it done in a rush. I hoped I could do better later. Another thing I want to mention is that I planned to show the current day on default for the showtime page but a lot of cinemas are closed these days and no showtime. I decided to pick up the mid of March as the default day. I hoped it would not make you feel confused. I spent a lot of time on this Phase. I hope I could do better in the next Phase.