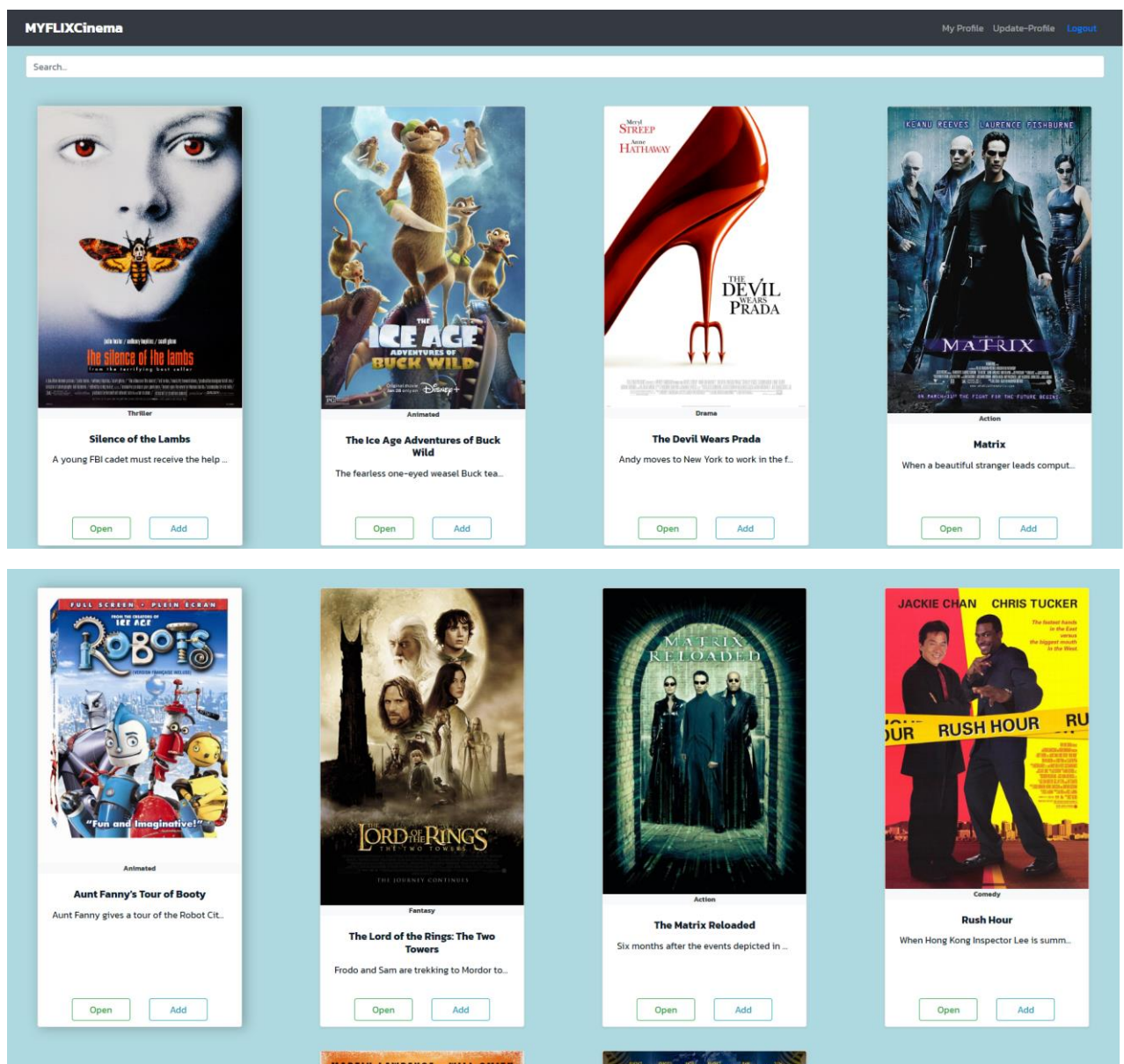


# Case Study for myFlix full-stack Project

## Overview

Nowadays the movie industry has been producing a lot, and after some time due to abundance of the movie products, we, users are not able to keep track on what movie we have experienced and when. Some users would like to keep track on their experience and keep their memories. myFlix app aims to provide to users information about movies to track their best moments and memories about the particular movie they watched.

myFlix is a web app, built with MERN (MongoDB, Express, React, Node.js) stack, that provides users with access to information about different movies, directors and genres. Users will be able to sign up, update their personal information, and create a list of their favourite movies.



## Purpose & Context

myFlix was a personal project, which was built as a part of my full-stack web development course at CareerFoundry to demonstrate my mastery of full-stack and JavaScript development.

During the development process, skills in Express, Node.js, MongoDB and React have been gained.

## Objective

The project is the first full stack project, which can showcase my first experience and gained knowledge in web development. I have faced few challenges in the project. First of all, the server-side or backend side was all about 'invisible logic' which was difficult to discern. The Client-side, or frontend side was all about discovering React and getting it work with server-side.

## Approach

## Server-side

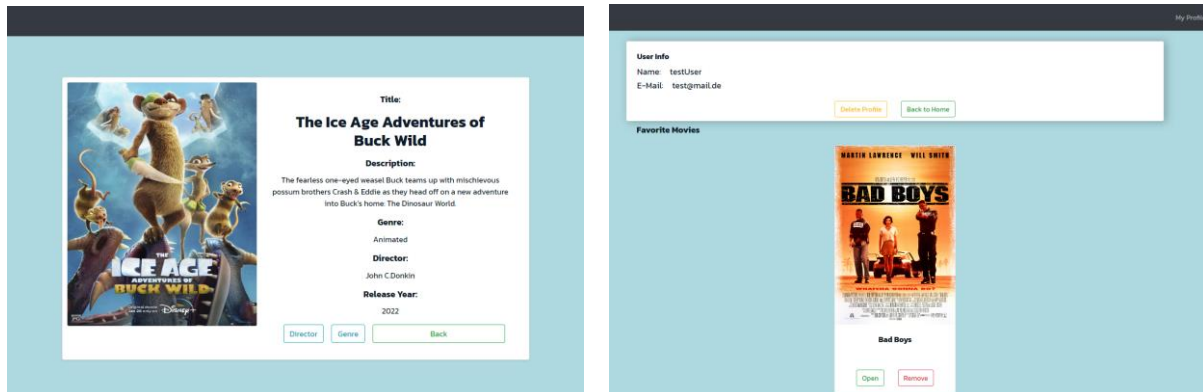
Server-side was developed with Node.js and Express.js. that was using MongoDB to build database. I created the API using REST architecture, with URL endpoints corresponding to each data operations and accessed with HTTP methods like GET, POST, DELETE and PUT. The data provided in JSON format.

For testing API Endpoints, I was using Postman. For security reasons authentication and authorization methods were applied. Check out the code here: [https://github.com/kraft-aka/movie\\_api](https://github.com/kraft-aka/movie_api)

```
D:\> IT > CF> careerfoundry > movie_api > {} movies.json > ...
1  {
2    "id": {
3      "id": "62594bc7eeb21d42108dd51b"
4    },
5    "Title": "Silence of the Lambs",
6    "Description": "A young FBI cadet must receive the help of an incarcerated and manipulative cannibal killer to help catch another serial killer.",
7    "Genre": {
8      "Name": "Thriller",
9      "Description": "Thriller film, also known as suspense film or suspense thriller, is a broad film genre that involves excitement and suspense in the audience."
10   },
11   "Director": {
12     "Name": "Jonathan Demme",
13     "Bio": "Robert Jonathan Demme was an American director, producer, and screenwriter.",
14     "Birth": "1944",
15     "Death": "2017"
16   },
17   "ImagePath": "https://m.media-amazon.com/images/M/MV5BNjhhZTk8ZmEhNjJhM108YzZlLWU1MmE1YzYzMjZmQW9hM2U4XkE5YXkFqGdeQXVyMjU0OTQ0OTY@_V1_.jpg",
18   "Featured": true,
19   "Actors": []
20 }
21
22 {
23   "id": {
24     "id": "62596bf7eeb21d42108dd51e"
25   },
26   "Title": "Matrix",
27   "Description": "When a beautiful stranger leads computer hacker Neo to a forbidding underworld, he discovers the shocking truth--the life he knows is the elaborate deception of an evil cyber-intelligence.",
28   "Genre": {
29     "Name": "Action",
30     "Description": "Action film is a film genre in which the protagonist is thrust into a series of events that typically involve violence and physical feats."
31   },
32   "Director": {
33     "Name": "Lana Wachowski",
34     "Bio": "Lana Wachowski and her sister Lilly Wachowski, also known as the Wachowskis, are the duo behind such ground-breaking movies as 'Matrix' (1999) and 'Cloud Atlas' (2012). Born to mother Lynne, a nurse, and father
35     "Birth": "1965",
36     "Death": ""
37   },
38   "ReleaseYear": "1999",
39   "ImagePath": "https://m.media-amazon.com/images/M/MV5BNzQzOTk3OTAtMDQ0ZmVklWl08TTEpD1lZjNkYzYzNjYjMjY2LWVkd1kEYXkFqGdeQXVyMjU0OTQ0OTY@_V1_FMjpg_UX1000_.jpg",
40   "Featured": true,
41   "Actors": [
42     {
43       "id": "62596f8deeb21d42108dd51e"
44     },
45     {
46       "id": "62597305eeb21d42108dd51f"
47     },
48     {
49       "id": "62596b2c7eeb21d42108dd51c"
50     },
51     {
52       "id": "62596b2c7eeb21d42108dd51c"
53     },
54     {
55       "id": "62596b2c7eeb21d42108dd51c"
56     },
57     {
58       "id": "62596b2c7eeb21d42108dd51c"
59     },
60     {
61       "id": "62596b2c7eeb21d42108dd51c"
62     },
63     {
64       "id": "62596b2c7eeb21d42108dd51c"
65     },
66     {
67       "id": "62596b2c7eeb21d42108dd51c"
68     },
69     {
70       "id": "62596b2c7eeb21d42108dd51c"
71     },
72     {
73       "id": "62596b2c7eeb21d42108dd51c"
74     },
75     {
76       "id": "62596b2c7eeb21d42108dd51c"
77     },
78     {
79       "id": "62596b2c7eeb21d42108dd51c"
80     },
81     {
82       "id": "62596b2c7eeb21d42108dd51c"
83     },
84     {
85       "id": "62596b2c7eeb21d42108dd51c"
86     },
87     {
88       "id": "62596b2c7eeb21d42108dd51c"
89     },
90     {
91       "id": "62596b2c7eeb21d42108dd51c"
92     },
93     {
94       "id": "62596b2c7eeb21d42108dd51c"
95     },
96     {
97       "id": "62596b2c7eeb21d42108dd51c"
98     },
99     {
100      "id": "62596b2c7eeb21d42108dd51c"
101    },
102    {
103      "id": "62596b2c7eeb21d42108dd51c"
104    },
105    {
106      "id": "62596b2c7eeb21d42108dd51c"
107    },
108    {
109      "id": "62596b2c7eeb21d42108dd51c"
110    },
111    {
112      "id": "62596b2c7eeb21d42108dd51c"
113    },
114    {
115      "id": "62596b2c7eeb21d42108dd51c"
116    },
117    {
118      "id": "62596b2c7eeb21d42108dd51c"
119    },
120    {
121      "id": "62596b2c7eeb21d42108dd51c"
122    },
123    {
124      "id": "62596b2c7eeb21d42108dd51c"
125    },
126    {
127      "id": "62596b2c7eeb21d42108dd51c"
128    },
129    {
130      "id": "62596b2c7eeb21d42108dd51c"
131    },
132    {
133      "id": "62596b2c7eeb21d42108dd51c"
134    },
135    {
136      "id": "62596b2c7eeb21d42108dd51c"
137    },
138    {
139      "id": "62596b2c7eeb21d42108dd51c"
140    },
141    {
142      "id": "62596b2c7eeb21d42108dd51c"
143    },
144    {
145      "id": "62596b2c7eeb21d42108dd51c"
146    },
147    {
148      "id": "62596b2c7eeb21d42108dd51c"
149    },
150    {
151      "id": "62596b2c7eeb21d42108dd51c"
152    },
153    {
154      "id": "62596b2c7eeb21d42108dd51c"
155    },
156    {
157      "id": "62596b2c7eeb21d42108dd51c"
158    },
159    {
160      "id": "62596b2c7eeb21d42108dd51c"
161    },
162    {
163      "id": "62596b2c7eeb21d42108dd51c"
164    },
165    {
166      "id": "62596b2c7eeb21d42108dd51c"
167    },
168    {
169      "id": "62596b2c7eeb21d42108dd51c"
170    },
171    {
172      "id": "62596b2c7eeb21d42108dd51c"
173    },
174    {
175      "id": "62596b2c7eeb21d42108dd51c"
176    },
177    {
178      "id": "62596b2c7eeb21d42108dd51c"
179    },
180    {
181      "id": "62596b2c7eeb21d42108dd51c"
182    },
183    {
184      "id": "62596b2c7eeb21d42108dd51c"
185    },
186    {
187      "id": "62596b2c7eeb21d42108dd51c"
188    },
189    {
190      "id": "62596b2c7eeb21d42108dd51c"
191    },
192    {
193      "id": "62596b2c7eeb21d42108dd51c"
194    },
195    {
196      "id": "62596b2c7eeb21d42108dd51c"
197    },
198    {
199      "id": "62596b2c7eeb21d42108dd51c"
200    },
201    {
202      "id": "62596b2c7eeb21d42108dd51c"
203    },
204    {
205      "id": "62596b2c7eeb21d42108dd51c"
206    },
207    {
208      "id": "62596b2c7eeb21d42108dd51c"
209    },
210    {
211      "id": "62596b2c7eeb21d42108dd51c"
212    },
213    {
214      "id": "62596b2c7eeb21d42108dd51c"
215    },
216    {
217      "id": "62596b2c7eeb21d42108dd51c"
218    },
219    {
220      "id": "62596b2c7eeb21d42108dd51c"
221    },
222    {
223      "id": "62596b2c7eeb21d42108dd51c"
224    },
225    {
226      "id": "62596b2c7eeb21d42108dd51c"
227    },
228    {
229      "id": "62596b2c7eeb21d42108dd51c"
230    },
231    {
232      "id": "62596b2c7eeb21d42108dd51c"
233    },
234    {
235      "id": "62596b2c7eeb21d42108dd51c"
236    },
237    {
238      "id": "62596b2c7eeb21d42108dd51c"
239    },
240    {
241      "id": "62596b2c7eeb21d42108dd51c"
242    },
243    {
244      "id": "62596b2c7eeb21d42108dd51c"
245    },
246    {
247      "id": "62596b2c7eeb21d42108dd51c"
248    },
249    {
250      "id": "62596b2c7eeb21d42108dd51c"
251    },
252    {
253      "id": "62596b2c7eeb21d42108dd51c"
254    },
255    {
256      "id": "62596b2c7eeb21d42108dd51c"
257    },
258    {
259      "id": "62596b2c7eeb21d42108dd51c"
260    },
261    {
262      "id": "62596b2c7eeb21d42108dd51c"
263    },
264    {
265      "id": "62596b2c7eeb21d42108dd51c"
266    },
267    {
```

## Client-side

After server-side was completed, I started working on client-side of the app. The popular library of JavaScript - React was used for that purpose. Client-side is a single page application (SPA), which is responsive and provides best user experience. It provides few interface views, including main view (displays list of all movies to logged in users), single movie view (displays the information about the movie and contains button to add the movie to favourite movies), login view, registration view and profile view (here user is able to update information such as username, password, email and list of favourite movies). Check out the code here: <https://github.com/kraft-aka/myFlix-client/tree/main/src>



## Challenges

This was my first time to build a server-side and client-side of the web application. It was a big challenge for me. I have enjoyed building a databases ( I learnt PostgreSQL and MongoDB) and working with terminal. The most challenging part from server-side was applying JWT token and passport. It took me some time to understand how they work.

```
10
11 // basic HTTP authentication for login requests
12 passport.use(new LocalStrategy({
13   usernameField: 'Username',
14   passwordField: 'Password'
15 }, (username, password, callback) => {
16   console.log(username + ' ' + password);
17   Users.findOne({ Username: username }, (error, user) => { // Mongoose used to check the username in DB
18     if (error) {
19       console.log(error);
20       return callback(error);
21     }
22
23     if (!user) {
24       console.log('incorrect username');
25       return callback(null, false, {message: 'Incorrect username.'});
26     }
27
28     if (!user.validatePassword(password)) {
29       console.log('incorrect password');
30       return callback(null, false, {message: 'Incorrect password.'});
31     }
32
33     console.log('finished');
34     return callback(null, user);
35   });
36 }));
```

Another challenge was understanding how the endpoints of API must be built and how to test them with Postman. The routes logic was defined in separate files (movie.js, users.js, actors.js) which allowed me to keep main index.js file clean, concise and easy to maintain.

Building the frontend part, it took me longer to understand how React and Redux work and how to achieve the desired results. For instance, I had to apply routing in React, which was seemed easy at first glance. But it was delusion, I understood it lately. The app was to

interact with backend endpoints and I was having some trouble to connect them, and get them work as they had to.

```
78     return (  
79       <Router>  
80         <Col md={12} style={{ padding: 0 }}>  
81           <MenuBar user={user} movies={movies} />  
82  
83           <Row className="main-view justify-content-md-center">  
84             <Route  
85               exact  
86               path="/"   
87               render={() => {  
88                 if (!user)  
89                   return (  
90                     <Col md={8}>  
91                       <LoginView onLoggedIn={(user) => this.onLoggedIn(user)} />  
92                     </Col>  
93                   );  
94                 }  
95               } />  
96             </Route>  
97           </Row>  
98         </Col>  
99       </Router>  
100     );  
101   }  
102 }  
103
```

But with the help of specific documentation, tutorials and help from my tutor and mentor I managed that task well.

The lessons I learnt during this project, are helping me now to build small web apps. However, I still work on enhancing my skills and knowledge working with MERN stack, specifically with React.

## Duration

The development of client-side took me 34 days to develop it. Whereas, the client-side took me a bit longer 49 days as I needed more time for React and Redux. Overall, I spent 83 days for this full-stack project.

## Credits

Role: Lead Developer

Tutor: Ignatius Sani

Mentor: Victor Miclovich