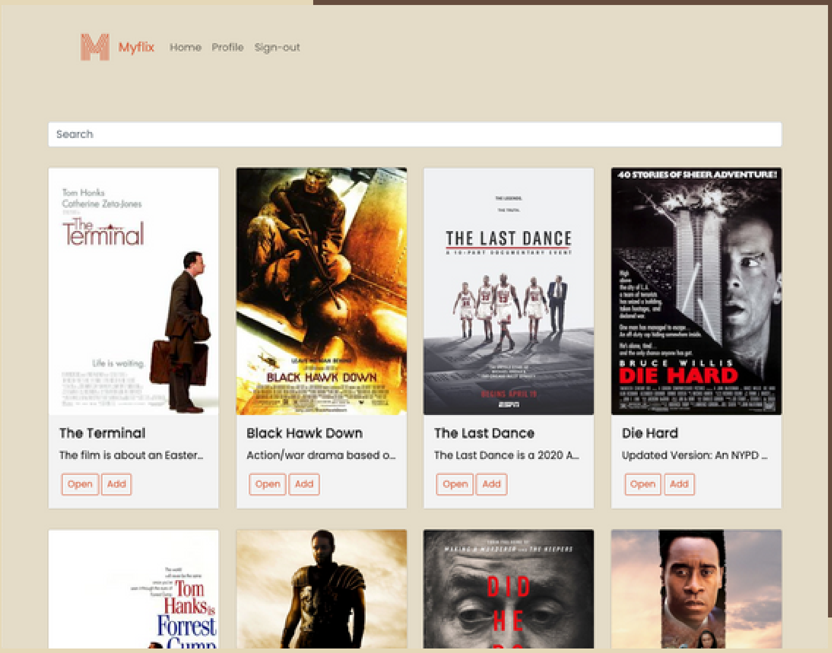
MYFLIX

CASE STUDY

EXERCISE 1.4


Myflix Home Profile Sign-out

Search

The Terminal
The film is about an Easter...
Open Add

Black Hawk Down
Action/war drama based o...
Open Add

The Last Dance
The Last Dance is a 2020 A...
Open Add

Die Hard
Updated Version: An NYPD ...
Open Add

AT A GLANCE

Challenges

- Learning on the go
- Time pressure
- Bug fixing

Learnings/Takeaways

- Web development takes time and it makes no sense to rush
- Making sure to understand core concepts in a fundamental way
- One of my strong points is the UI design part

"myFlix has been my favourite but also most challenging and time consuming project throughout this program. While the server-side part, was easier to understand, I had more fun with the client-side, especially the UI design part."

Patrick Lemmer

Conclusion

OVERVIEW

A web app providing users with access to information about different movies, directors, and genres. Users will also be able to sign up, update their personal information, and create a list of their favourite movies.

CONTEXT

The project was built as part of my Full-Stack Web Development course at CareerFoundry to demonstrate my current skill set in Full-Stack JavaScript Development.

OBJECTIVE

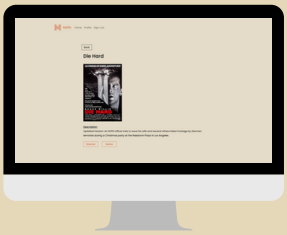
The aim of the project was to have an ambitious full-stack project that I can add to my professional portfolio.

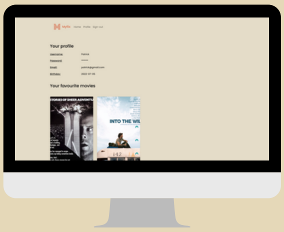
PROJECT TIMELINE

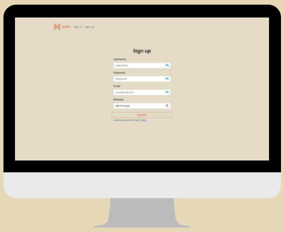
The project was concluded in July 2022 after a development period of a total of five weeks.

CREDITS

Developer: Patrick Lemmer
Tutor: Bless Darah
Mentor: Ramadhan Aheebwa







↓ Read on, for a detailed breakdown ↓

APPROACH

The whole project was built in two steps:

1. Server-side
2. Client-side

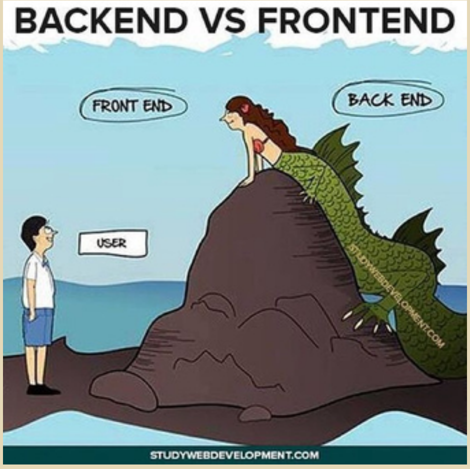
Both parts have a step description describing the deliverables based on the program's curriculum and some thoughts and notes of mine to add my personal experiences.

One important side note: The project was part of my career change program with CareerFoundry. The whole program consisted of several achievements, each with a different focus, but still based on each other. Since the server-side part of the myFlix project was built first, I did not know about the client-side part in detail at that point.

PART 1 - SERVER-SIDE

Step 1: The very first thing I did was go through the project brief provided by CareerFoundry, in order to get an understanding of the project's objective and its requirements.

Looking back, studying this project brief was definitely a crucial step, in order for me to understand the overall architecture of a full-stack application, like myFlix. I already knew the differences between server-side and client-side, but lacked an understanding of how both parts work together in a technical way. I also referred back to the project brief throughout the entire length of the myFlix project.



Step 2: Before getting into server-side related tools and concepts, I learned about the difference between server-side and client-side in more detail.

The differences were elaborated on based on the example of a supermarket, in which everything happening behind the scenes (not visible to the customer) would be referred to as server-side and everything visible to the customer would be referred to as client-side. This analogy wasn't entirely new to me, but it really helped to stop and think about what this actually means in the field of full-stack web development. At this point it was very clear to me that I will have to go through the following steps with a good level of awareness for detail in order to be able to connect the dots when starting with the client-side.

Step 3: This project was about retrieving data from a database. Therefore it was now time to understand what an (RESTful) API is, how it works and how a user would be able to access the data through the client-side.

Like with some other concepts I learned throughout the entire program, I already had a basic understanding of what an API is and how it works. However, at this point it was incredibly helpful for me to have a more detailed explanation of APIs and specifically in relation to my project. The building process of the API happened “on the go” throughout this achievement.

Step 4: In order to start building the server-side part of myFlix, I then learned how to use Node.js, its “building blocks”, Node modules, and how to navigate the terminal (macOS).

At this point, this project started to get a bit more complicated for me. There was a lot of confusion, mostly due to all the unknown (module) names, descriptions, dependencies and so on. I had difficulties connecting dots and fitting some of the things I got introduced to into the bigger picture. Looking back now, I am certain that this was related to the programs curriculum. Due to my personal progress, which was slower than expected at times, there was quite some time between the single exercises of the achievement. For this reason, I sometimes struggled to remember how and why specific things were connected. On the plus side however, I managed to navigate the terminal rather quickly.

Step 5: Since modules are so important, it was also crucial to know how they can be managed, organized, updated etc. with the help of the package manager npm and also that those packages become dependencies in the context of a project.

The issues mentioned in the previous step, were alleviated a little bit, with the introduction of npm into this project. The concept of dependencies, gave me some much needed clarity, overview and structure. I also gained back some confidence, once I understood the concept of the "package.json" file and the differences between global and local installation of packages.

Step 6: For this project, a server-side framework was used, which meant that I had to get to know and set up Express in the next step. I also got introduced to Middleware and Express error handling at this point.

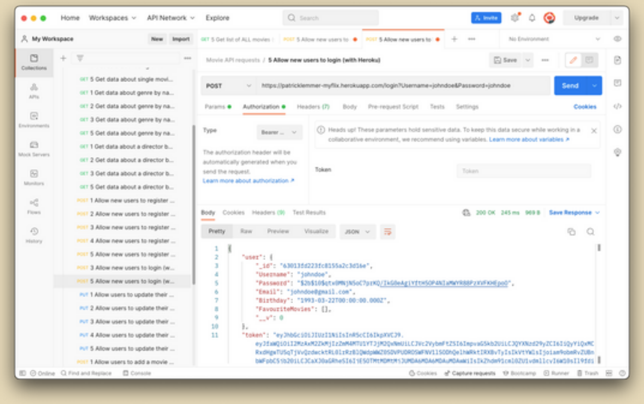
Once again, the introduction of Express and Middleware made me slow down a bit again. I needed to make sure that I understood the context and the bigger picture, in order to move on without too much uncertainty.

Step 7: It was then time to think about the design of the REST API for this project. This meant to get familiar with HTTP requests and CRUD operations.

Learning about HTTP requests and CRUD was pretty straight forward (for me), since there are clear definitions behind these concepts. Again, I could make up some time at this point. Looking back, I feel like "waves" describe my learning experience throughout the program best. Some parts took me longer to get a good understanding of, and other parts came more natural to me. This behaviour is pretty common I believe, but this project (program) made me aware of what I personally need in order to have a consistent learning experience.

Step 8: Once the endpoints for myFlix had been defined and the overall architecture had been done, it was time to test the API using Postman and create meaningful documentation.

Testing my endpoints in Postman, was one of a few confident boosters for me. Initially I did have to fix a few things in order to make all endpoints pass the tests, but by doing that and then getting the immediate confirmation of a test being successful really helped me at that time. I just knew that I created something from scratch and made it work as well. For a new developer, it's gold to experience these small success stories.



GET 5 Get list of ALL movies
POST 5 Allow new users to
POST 5 Allow new users to

GET 5 Get data about single movi...
GET 1 Get data about genre by na...
GET 2 Get data about genre by na...
GET 3 Get data about genre by na...
GET 5 Get data about genre by na...
GET 1 Get data about a director b...
GET 2 Get data about a director b...
GET 3 Get data about a director b...
GET 5 Get data about a director b...
POST 1 Allow new users to register ...
POST 2 Allow new users to register ...
POST 3 Allow new users to register ...
POST 4 Allow new users to register ...
POST 5 Allow new users to register ...
POST 3 Allow new users to login (w...
POST 5 Allow new users to login (w...
PUT 1 Allow users to update their ...
PUT 2 Allow users to update their ...
PUT 3 Allow users to update their ...
PUT 4 Allow users to update their ...
PUT 5 Allow users to update their ...
POST 1 Allow users to add a movie ...

Movie API requests / 5 Allow new users to login (with Heroku)

POST https://patricklemmer-myflix.herokuapp.com/login?Username=johndoe&Password=johndoe

Type: Bearer ...

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

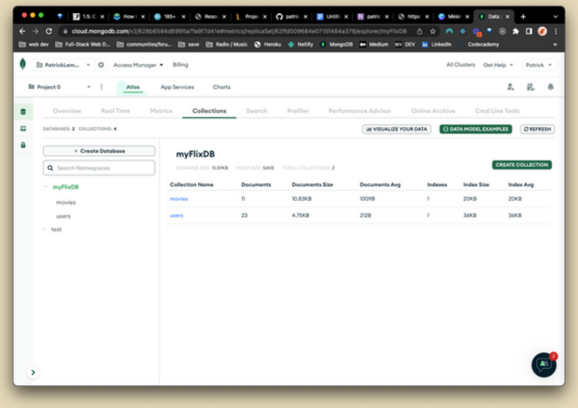
The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

200 OK 245 ms 969 B

```
{
    "user": {
        "_id": "63013fd223fc8155a2c3d16e",
        "Username": "johndoe",
        "Password": "$2b$10$qtw0MNjN5oC7prKQ/IkG0eAgiYftH5OP4NIaMWYR8BPzWVFKHEpoO",
        "Email": "johndoe@gmail.com",
        "Birthday": "1993-03-22T00:00:00.000Z",
        "FavouriteMovies": [],
        "__v": 0
    },
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJfaWQiOiI2MzAxM2ZkMjIzZmM4MTU1YTJjM2QxNmUiLCJVc2VybmFtZSI6ImpvaG5kb2UiLCJQYXNzd29yZCI6IiQyYiQxMC
        [illegible]
        [illegible]
```

Step 9: In the next step I got introduced to the concept of relational and non-relational databases and SQL.

Once the endpoints in the previous step had been created successfully, it was time to find a data storage solution. In my opinion, the program does well by teaching not only the solutions, necessary for a project, but also some alternatives for other (later) projects. PostgreSQL was such an alternative in this case and I therefore took some time to learn some basics before moving on with this projects non-relational database solution, MongoDB, in the next step. This part, was another part of this course, which I made sense of fairly quick and therefore had quite some fun in the process of setting up MongoDB for my project.



cloud.mongodb.com/v2/628b6584d6995a7fa9f7d41e#metrics/replicaSet/62ffd009684e0715f484a378/explorer/myFlixDB

DATABASES: 2 COLLECTIONS: 4

myFlixDB

DATABASE SIZE: 15.59KB INDEX SIZE: 56KB TOTAL COLLECTIONS: 2

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
movies	11	10.83KB	1009B	1	20KB	20KB
users	23	4.75KB	212B	1	36KB	36KB

Step 10: Now, I was ready to integrate the newly created database with my API using business logic.

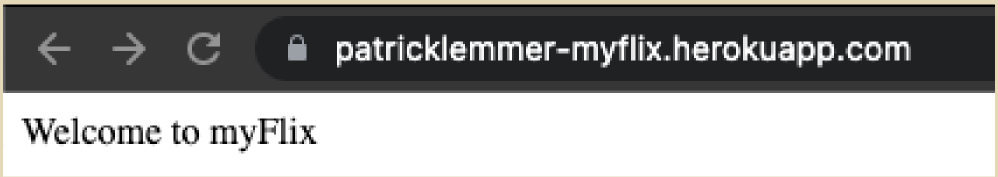
In this step, I used Mongoose to write data models with Node and Express. For this project, that meant to first define what kind of data is used and then create models based on that. As there were only "users" and "movies" this process was fairly uncomplicated and offered a great introduction into the whole topic of data modelling and working with data in general.

Step 11: Next, I learned different ways of server-side authentication and authorization before adding both to the myFlix project.

See step 8 of the client-side part

Step 12: In the last step I modified my API to align it with some data security regulations. Once that was done, I was ready to deploy the API to Heroku.

I got up to this point with some bugs and "a normal amount" of challenges along the way, which I solved with my tutor and mentor as sparring partners. However, after implementing the necessary security requirements and deploying to Heroku, I had to come back to this achievement at a later point and start again from scratch. It turned out, that my connection between Heroku and MongoDB caused some issues, which would only become apparent during the development phase of the client-side part. After some rounds of bug fixing together with my tutor and mentor, I decided to go back to the start of this achievement, and write the server-side code again. I made that decision, mainly due to the time pressure I faced and the knowledge that I would likely be able to get the client-side code ready for re-deployment in a reasonable amount of time.



Welcome to myFlix

PART 2 - CLIENT-SIDE

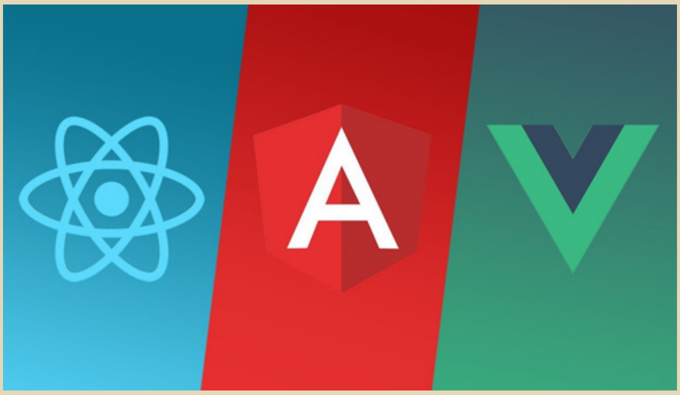
Step 1: Similar to the server-side part, I started off with studying the project brief in order to get familiar with the requirements for the client-side part.

Studying the project brief has turned out to be a very helpful step in the beginning of every project. As this is a standard part of workflows in most organizations, I also tried to get as much experience in reading, understanding and then executing the given information in these briefs.

Step 2: Before diving into frameworks and libraries in the next step, I got introduced to JavaScript design patterns and as a consequence I learned about the MVC architecture, which was applied in this client-side project.

Back when I was going through this achievement for the first time, I wasn't aware of the topic of design patterns such as the MVC architecture in web development. I always connected design patterns with the visual appearance of a website. I was therefore not able to move on before I felt I had a solid understanding of this topic. It also turned out in the next step, that this knowledge was indeed necessary for me to move on and make any informed decisions about a fitting framework for my project. I can now say, that I am happy to have taken more time for some of the tasks and the concepts behind them, because there are many core concepts throughout this program, that inevitably need to be understood to make sense of everything else.

Step 3: While I already knew about the general idea of frameworks and libraries at this point, I now had to spend some time to understand why and how to choose the right framework for a project, and more importantly for the myFlix project.



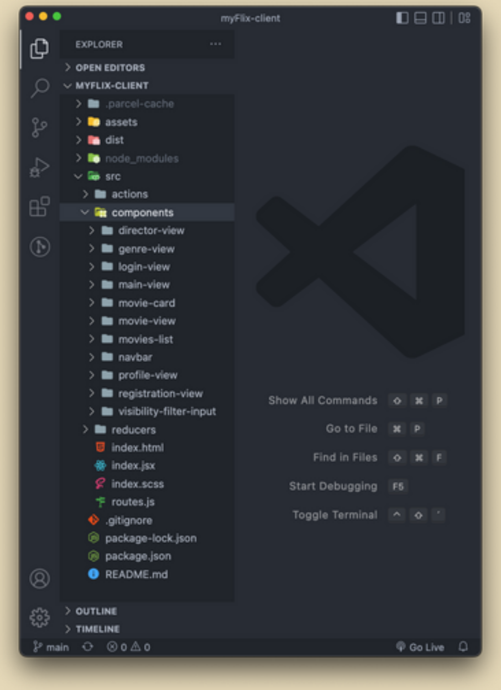
Even though the decision of which framework I would choose was made for me by the curriculum of the program, I took some time to learn the differences between React, Angular and Vue and why React would be a good fit for this project. For me, as a new developer and at that time, React immediately seemed to be the most attractive solution as well. I am aware however, that each of these frameworks have their benefits and "right to exist".

Step 4: Once the decision was made to use React for the client-side of the myFlix app, it was time for me to get familiar with it and its basics first.

Learning the basics of React (and later going into some more advanced React) was one of the bigger challenges throughout this achievement.

Step 5: In an earlier achievement of the program I already touched on some parts of the build process. However, at this point I learned about this process in more detail and got introduced to tools that take care of this process for me altogether. After deciding on Parcel as a build tool, I took some time to set it up for the myFlix project.

Going through this process in more detail, was very helpful to understand some core concepts that are essential to know as a web developer. Although there are tools, such as Parcel, that take care of the build process, it always makes sense to go through this process without such tools. For me personally, doing exactly that proved to be the most effective to grasp specific concepts. Knowing that I would be able to go through this process with and without those tools, gave me another confidence boost.



Step 6: At this point it was time for a deep dive into React and its core concepts in order to understand and build the architecture of this project.

Working against time pressure, was one of the biggest challenges throughout the entire program. Especially, when I reached some major topics such as React, I felt that it would not have the time I need in order to get a solid understanding of a given topic. It was therefore often a matter of making sure to learn and understand as much as possible to go through the achievements and knowing that I would definitely take the time to become more familiar with React at a later stage.

Step 7: Once the general architecture had been implemented, I first learned about Bootstrap and then integrated it into my project.

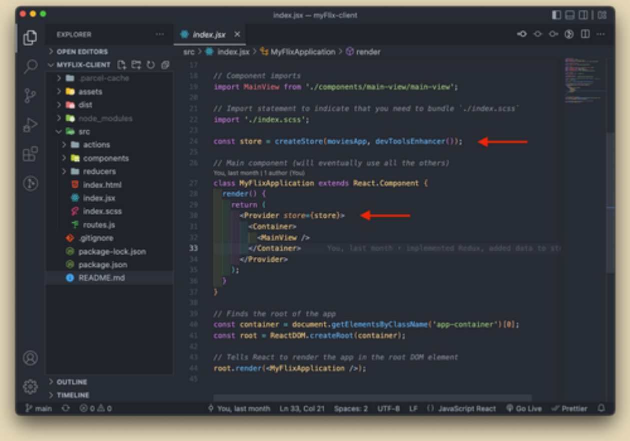
Not only do I feel that UI design is one of my stronger skills, I also got feedback from my tutor and mentor confirming this feeling of mine. Integrating Bootstrap into my project was incredibly fun and a way to build back some of the momentum I sometimes lost during more difficult phases. Also, I am a big fan of structure, organisation and making sure to work as effective as possible. UI libraries like Bootstrap make all of this possible, which is the reason I did enjoy working with it.

Step 8: In the next step, I learned about client-side authentication, the differences to server-side authentication and how to add it to the project.

The authentication topics for the server-side and client-side parts added some depth to my knowledge. Topics like this add sophistication to an application, and it feels good to know that I am able to achieve this level of sophistication already.

Step 9: The final step in this project was the implementation of React Redux, after taking some time to understand the concept and how it can be implemented into an existing React app.

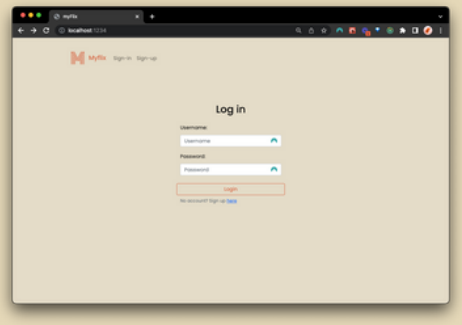
In the last part of this achievement and also of this project, I needed to add Redux to my React application. It was also the part where I spend the most amount of time, since Redux was implemented into my existing project, which was quite a challenge for me.

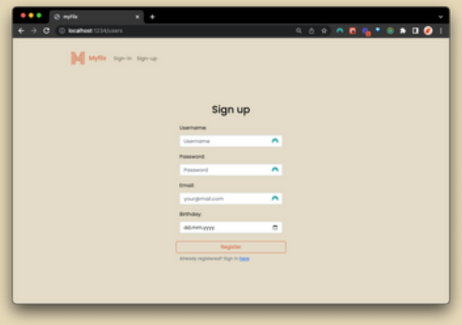

index.jsx — myFlix-client
EXPLORER
OPEN EDITORS
MYFLIX-CLIENT
.parcel-cache
assets
dist
node_modules
src
actions
components
reducers
index.html
index.jsx
index.scss
routes.js
.gitignore
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
index.jsx
src > index.jsx > MyFlixApplication > render
You, last month | 1 author (You)
You, last month • implemented Redux, added data to st
main
You, last month Ln 33, Col 21 Spaces: 2 UTF-8 LF {} JavaScript React Go Live Prettier

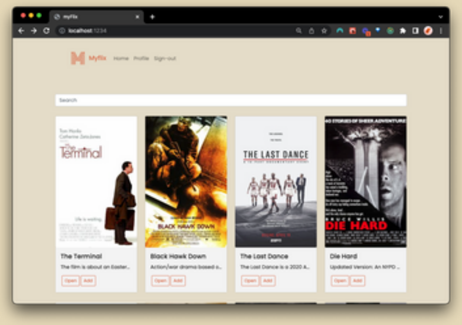
```
// Component imports
import MainView from './components/main-view/main-view';

// Import statement to indicate that you need to bundle `./index.scss`
import './index.scss';

const store = createStore(moviesApp, devToolsEnhancer());

// Main component (will eventually use all the others)
class MyFlixApplication extends React.Component {
  render() {
    return (
      <Provider store={store}>
        <Container>
          <MainView />
        </Container>
      </Provider>
    );
  }
}

// Finds the root of the app
const container = document.getElementsByClassName('app-container')[0];
const root = ReactDOM.createRoot(container);

// Tells React to render the app in the root DOM element
root.render(<MyFlixApplication />);
```

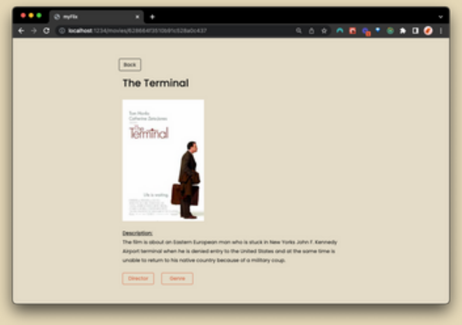
IMAGERY

A selection of screenshots from the myFlix movie database.

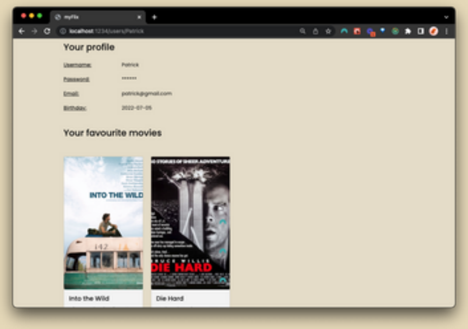




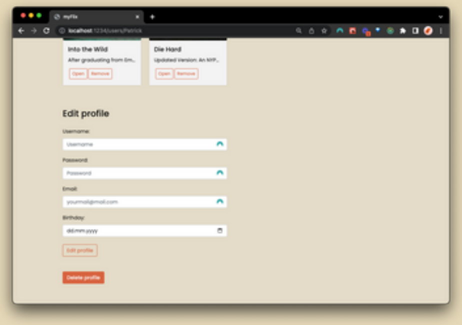

myflix
localhost:1234
Myflix
Home
Profile
Sign-out
Search
The Terminal
The film is about an Easter...
Black Hawk Down
Action/war drama based o...
The Last Dance
The Last Dance is a 2020 A...
Die Hard
Updated Version: An NYPD ...
Open
Add



The film is about an Eastern European man who is stuck in New Yorks John F. Kennedy Airport terminal when he is denied entry to the United States and at the same time is unable to return to his native country because of a military coup.



Username: Patrick
Password: ******
Email: patrick@gmail.com
Birthday: 2022-07-05


myflix
localhost:1234/users/Patrick
Into the Wild
After graduating from Em...
Die Hard
Updated Version: An NYP...
Open
Remove
Edit profile
Username:
Username
Password:
Password
Email:
yourmail@mail.com
Birthday:
dd.mm.yyyy
Edit profile
Delete profile