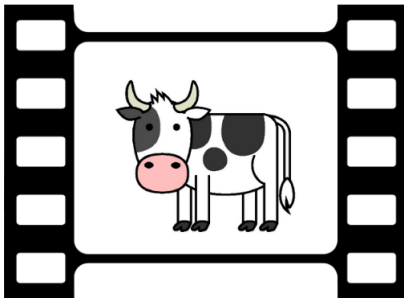
Case Study for Mooovies full-stack project



Overview
I created a film database and corresponding front-end client that would serve users data about the films in the database in a variety of ways, including information about the directors and genres associated with each film. It allows users to browse titles, create profiles, securely log in and out, edit their personal info, and add films to their list of favorites.

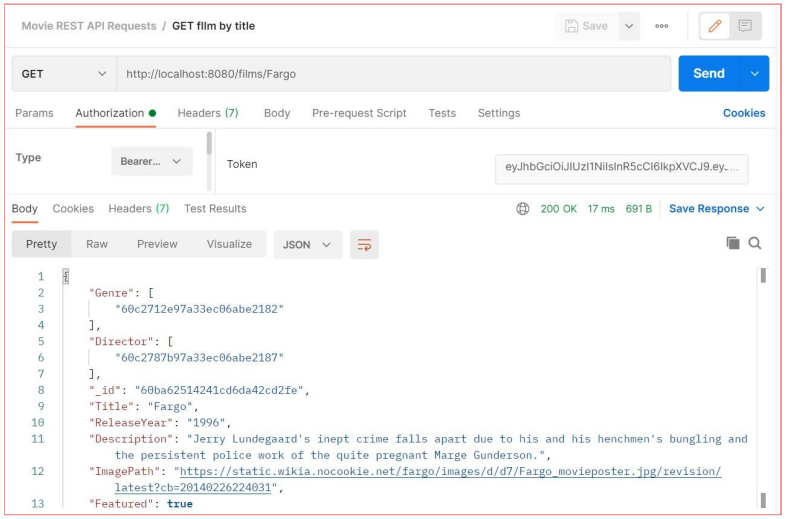
What?
The Mooovies app is built with Node.js, Express, MongoDB, and it has two front-end iterations, one built in React and the other in Angular.

Why?
This project was part of the CareerFoundry Full-Stack Immersion program, and it provided a fun way to learn and master all the elements that comprise MERN and MEAN stacks. It has also given me a cool and comprehensive app that I now feature on my portfolio website at www.lankyjoe.com.

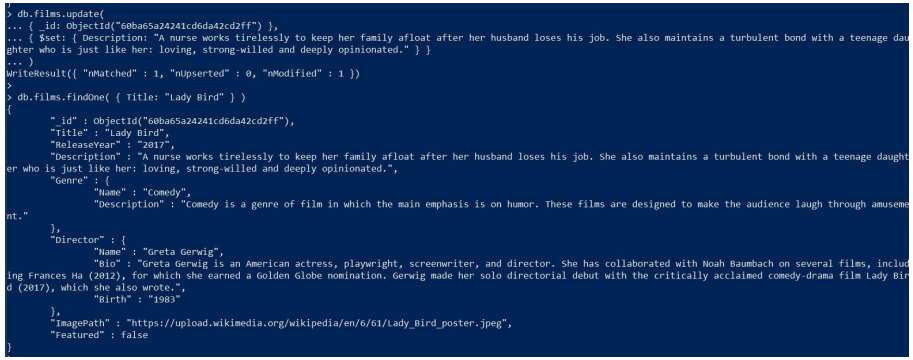
How?
I built this app “from scratch,” from the back end (server-side) to the two different front end (client-side) versions.

Back End

To begin with, I created a web server and RESTful API with Node.js and the Express framework. The API handles CRUD-style HTTP requests through URL endpoints. I then used Postman to test those endpoints.



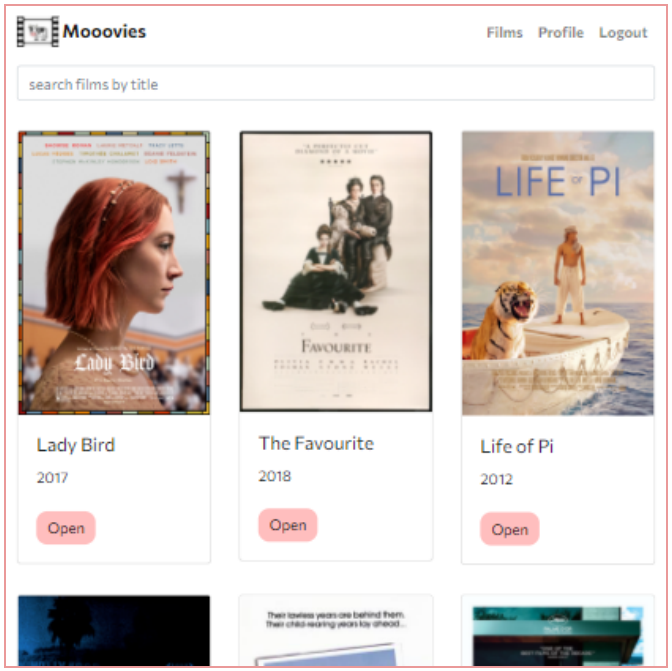
The app also uses basic HTTP authentication and authorization that is JWT/token-based. The authentication logic is also built into the URL endpoints.



I built a non-relational database with MongoDB and used Mongoose to link up the business logic and define the database schema. The database includes separate collections that refer to one another through primary keys and are included in read operations through the .populate method.

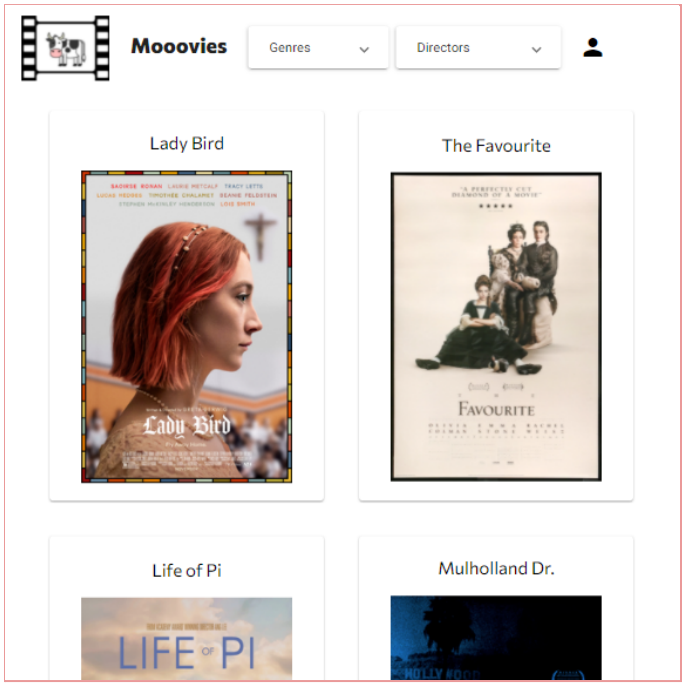
```
/**
 * Get the data about films of one genre from 'Films' collection
 * @method GET
 * @method populate adds data from Genre and Director documents
 * @param {string} endpoint - /Genre/:_id
 * @param {string} _id of genre
 * @requires authentication JWT
 * @returns {array} returns array of film objects in json format
 */
app.get('/Genre/:_id', passport.authenticate('jwt', { session: false }), (req, res) => {
  Films.find({ Genre: mongoose.Types.ObjectId(req.params._id) })
  .populate({path: 'Genre', model: Genres})
  .populate({path: 'Director', model: Directors})
    .then((films) => {
      res.json(films);
    })
    .catch((err) => {
      console.error(err);
      res.status(500).send('Error: ' + err);
    });
});
```

Front End



React version

Initially the frontend for this project was made using React and React-Redux. It uses React-Router to handle URL-based routing and is styled with help from React-Bootstrap. It features registration and login pages, a searchable main view that shows all the films in the database, individual film views that provide more details, and director and genre views that provide more information about those, including representative films. Users can also edit their profile information and add or remove films from a list of their favorites.



Angular version

Then it came time to learn Angular, and I built a slightly different version of the client-side app written in TypeScript for Angular. This version has almost all of the same features as the React version, and includes a pair of dropdown menus through which users can see a list of all directors and genres featured in the database, linking to pages with details about each. Styling is built based on Angular Material. Both client-side iterations of the Mooovies app are designed to be responsive on a variety of browsers and devices.

Challenges:
This project was full of challenges! Learning to use these (formerly) unfamiliar technologies was the most difficult aspect. It also took some time to carefully create and edit the database using the command terminal. Because the CareerFoundry program doesn't spoon-feed students with instructions for how to code their projects, I had to figure out a lot on my own, through internet research, and from colleagues familiar with the project. There was a ton of troubleshooting!

Duration:
This entire project, including both front-end iterations, took approximately three months to complete. If that seems like a long time, it's because I was learning to use all of the tools involved for the first time, with only some HTML, CSS, and a smattering of JavaScript under my belt to begin with. React took some time to get used to, and Angular was significantly more challenging.

Credits:
Developer: Joe Jordan
Tutor: Jason Early
Mentor: John Akhilomen

Links:

API code: https://github.com/shplank/movie_api

React code: https://github.com/shplank/Mooovies-client

Angular code: https://github.com/shplank/Mooovies-Angular

Finished Angular site: https://shplank.github.io/Mooovies-Angular/