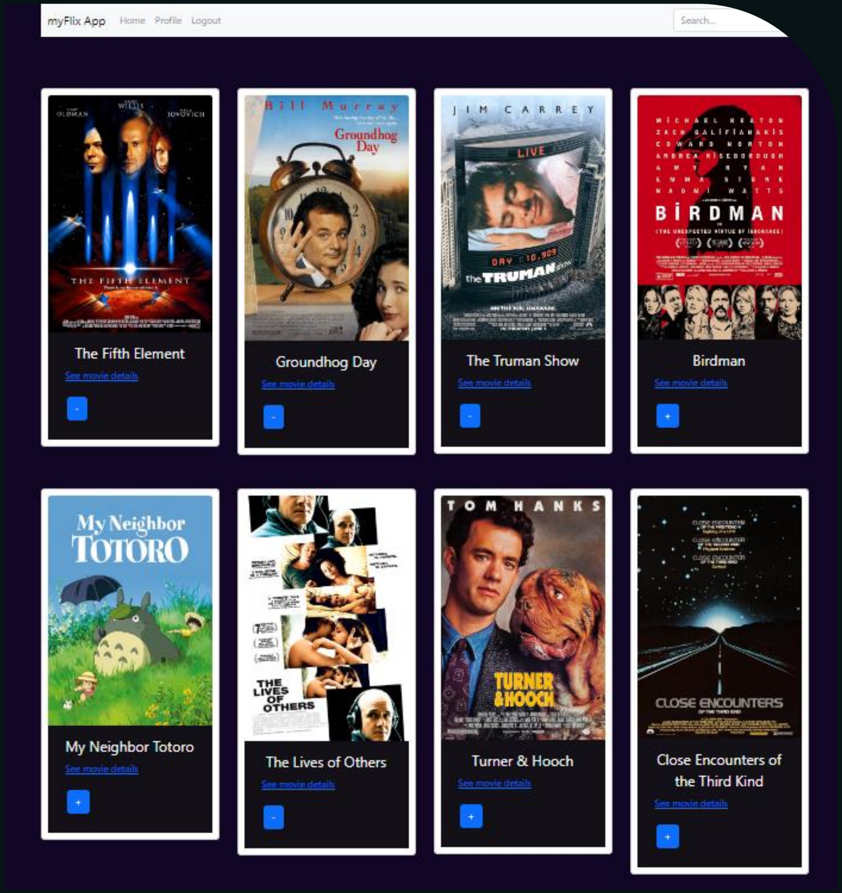
Full Stack Development

Case Study

Anca Wolf

myFlix App

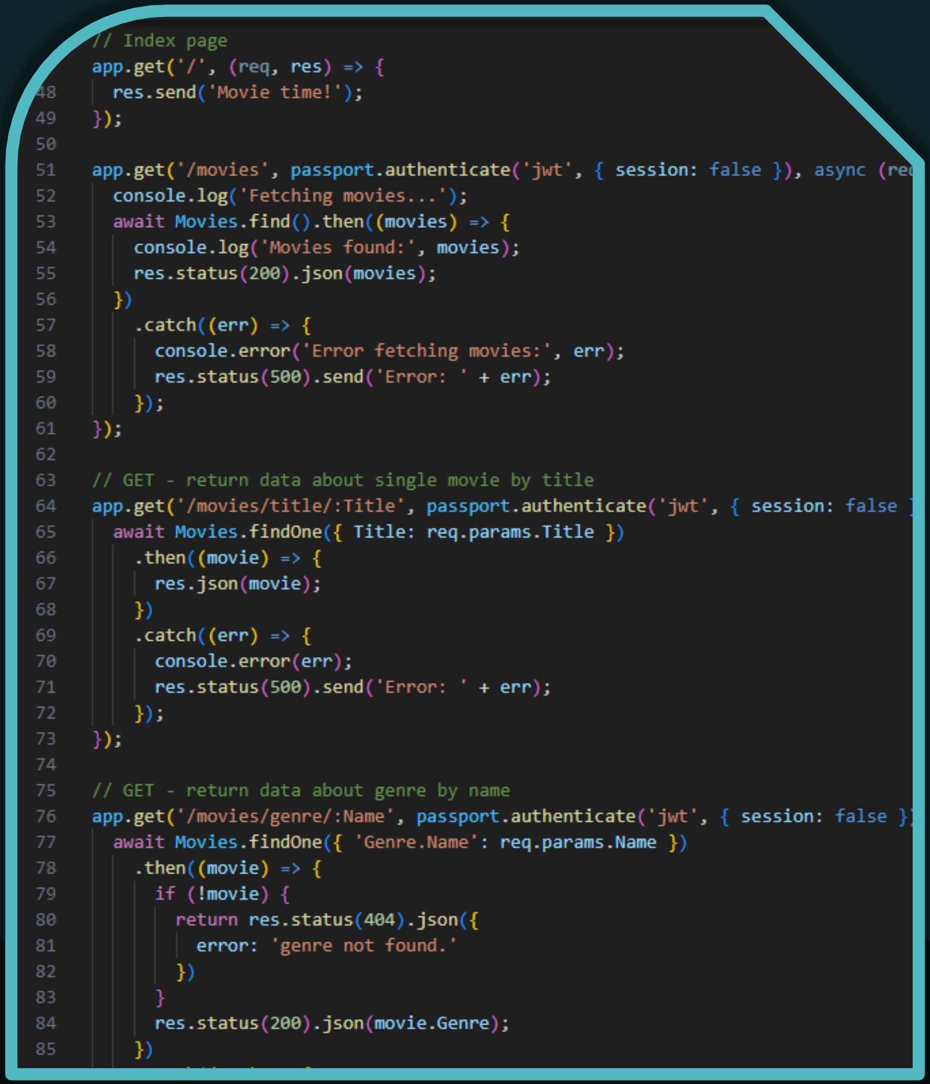


Overview

This is a web application constructed of a server-side and a client-side. Users will be able to sign up, update their profile information, access information about different movies, directors and genres, and create a list of their favourite movies.

Purpose

The creation of this project as part of my web development course at CareerFoundry aids to showcase the acquired expertise with the MERN stack (MongoDB, Express, React and Node.js).



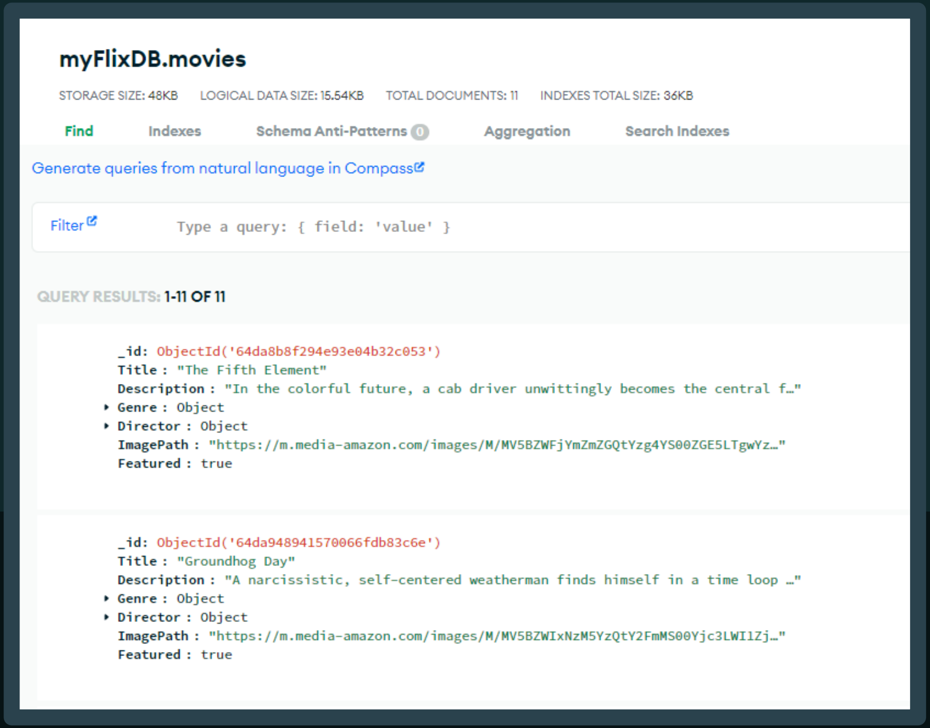
```
// Index page
app.get('/', (req, res) => {
  res.send('Movie time!');
});

app.get('/movies', passport.authenticate('jwt', { session: false }), async (re
  console.log('Fetching movies...');
  await Movies.find().then((movies) => {
    console.log('Movies found:', movies);
    res.status(200).json(movies);
  })
    .catch((err) => {
      console.error('Error fetching movies:', err);
      res.status(500).send('Error: ' + err);
    });
});

// GET - return data about single movie by title
app.get('/movies/title/:Title', passport.authenticate('jwt', { session: false }
  await Movies.findOne({ Title: req.params.Title })
    .then((movie) => {
      res.json(movie);
    })
    .catch((err) => {
      console.error(err);
      res.status(500).send('Error: ' + err);
    });
});

// GET - return data about genre by name
app.get('/movies/genre/:Name', passport.authenticate('jwt', { session: false })
  await Movies.findOne({ 'Genre.Name': req.params.Name })
    .then((movie) => {
      if (!movie) {
        return res.status(404).json({
          error: 'genre not found.'
        })
      }
      res.status(200).json(movie.Genre);
    })
```

Duration

Client-side and server-side development of this web application, from start to finish, extended over several months, during which I had the opportunity to get a better grasp of the technologies used.

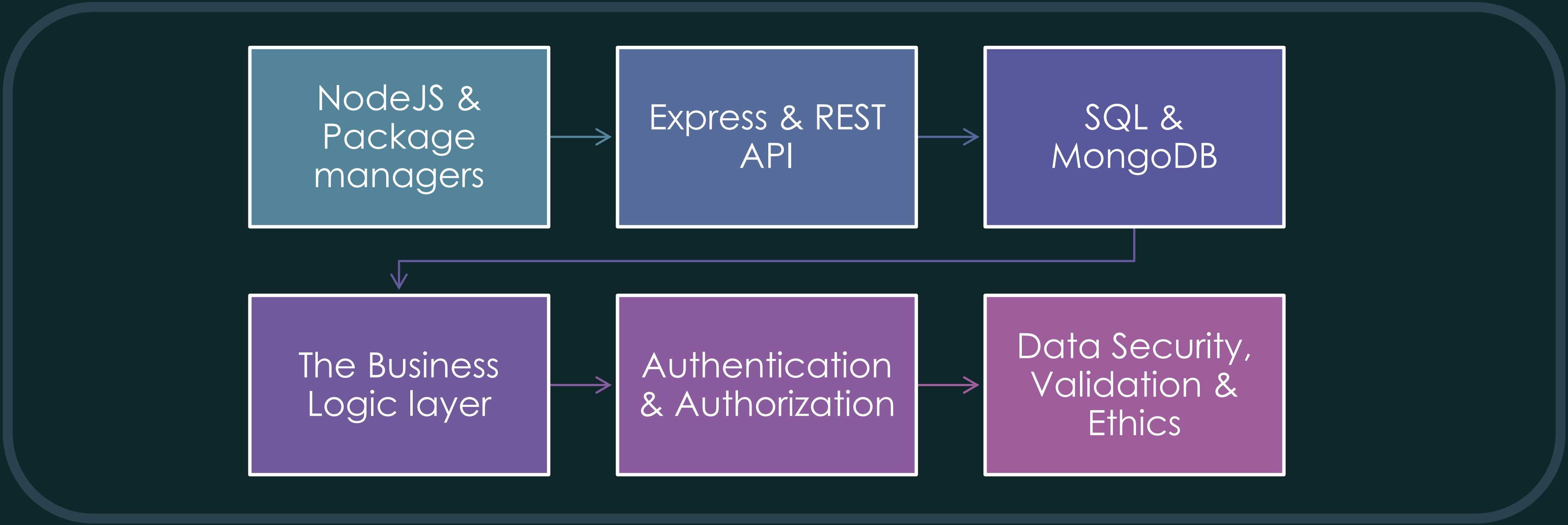
Methodologies

- JavaScript
- MERN Stack
- Postman
- Netlify
- Render
- Parcel
- Mongoose



```
  _id: ObjectId('64da8b8f294e93e04b32c053')
  Title : "The Fifth Element"
  Description : "In the colorful future, a cab driver unwittingly becomes the central f…"
▸ Genre : Object
▸ Director : Object
  ImagePath : "https://m.media-amazon.com/images/M/MV5BZWFjYmZmZGQtYzg4YS00ZGE5LTgwYz…"
  Featured : true

  _id: ObjectId('64da948941570066fdb83c6e')
  Title : "Groundhog Day"
  Description : "A narcissistic, self-centered weatherman finds himself in a time loop …"
▸ Genre : Object
▸ Director : Object
  ImagePath : "https://m.media-amazon.com/images/M/MV5BZWIxNzM5YzQtY2FmMS00Yjc3LWI1Zj…"
  Featured : true
```

Server side development

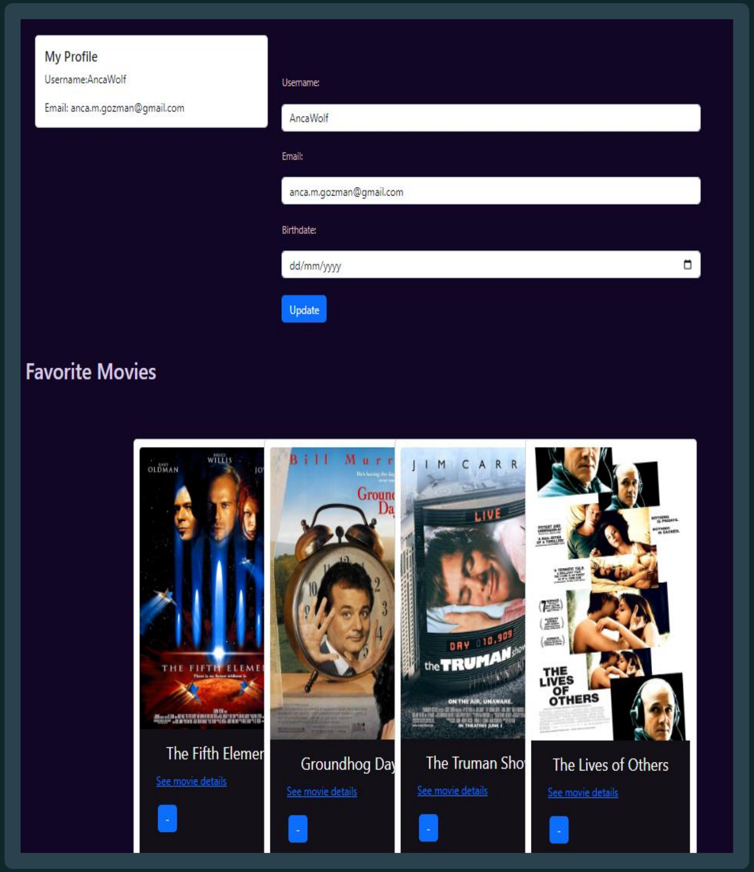

NodeJS & Package managers
Express & REST API
SQL & MongoDB
The Business Logic layer
Authentication & Authorization
Data Security, Validation & Ethics

Starting with user stories and essential features, the first packages were created, followed by URL endpoints corresponding to the data. After creating a relational database (SQL) with the use of PostgreSQL, the decision was made to continue with MongoDB, a non-relational database. Towards the finalization of the backend part, authorization and authentication was implemented with HTTP and JWT authentication.

Client side development

The second part of the project consisted of building the user interface, the client side, where users would be able to register an account and access the website’s contents. Developed with React and Parcel.

Afterwards the components for different views were created and the design of the general layout, as well as single-movie visualization, was added with the use of React Bootstrap library.



My Profile

Username:AncaWolf

Email: anca.m.gozman@gmail.com

Username: AncaWolf

Email: anca.m.gozman@gmail.com

Birthdate: dd/mm/yyyy

Favorite Movies

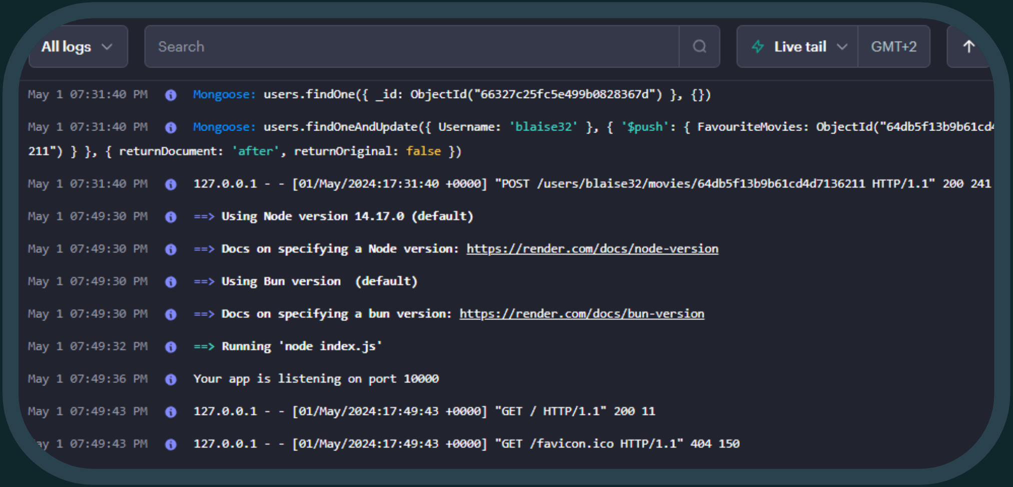
The Fifth Elemen

Groundhog Day

The Truman Sho

The Lives of Others

Challenges & Final Thoughts



```
May 1 07:31:40 PM  Mongoose: users.findOne({ _id: ObjectId("66327c25fc5e499b0828367d") }, {})
May 1 07:31:40 PM  Mongoose: users.findOneAndUpdate({ Username: 'blaise32' }, { '$push': { FavouriteMovies: ObjectId("64db5f13b9b61cd
211") } }, { returnDocument: 'after', returnOriginal: false })
May 1 07:31:40 PM  127.0.0.1 - - [01/May/2024:17:31:40 +0000] "POST /users/blaise32/movies/64db5f13b9b61cd4d7136211 HTTP/1.1" 200 241
May 1 07:49:30 PM  ==> Using Node version 14.17.0 (default)
May 1 07:49:30 PM  ==> Docs on specifying a Node version: https://render.com/docs/node-version
May 1 07:49:30 PM  ==> Using Bun version  (default)
May 1 07:49:30 PM  ==> Docs on specifying a bun version: https://render.com/docs/bun-version
May 1 07:49:32 PM  ==> Running 'node index.js'
May 1 07:49:36 PM  Your app is listening on port 10000
May 1 07:49:43 PM  127.0.0.1 - - [01/May/2024:17:49:43 +0000] "GET / HTTP/1.1" 200 11
ay 1 07:49:43 PM  127.0.0.1 - - [01/May/2024:17:49:43 +0000] "GET /favicon.ico HTTP/1.1" 404 150
```

Throughout the development process of this web application challenges were not missing. From discrepancies and incompatibilities between dependencies version, typos and faulty connections between the server-side and the client-side components, this project offered great opportunities to research in detail various behaviours and solutions, which led to a better understanding of the technologies used.

Overall, I am proud of the final product and glad to have gained a deeper insight into React.

Thank you!