MERN

Overview:
AppForMovies is a full stack web application that provides information about movies using the MERN Stack (MongoDB, Express, React and Node.js), the app allows users to create an account, update their personal data and create a list of their favorite movies.

Purpose & Context:
I've built the app as a part of my portfolio that I've completed in the CareerFoundry Full-Stack Web Developer course in order to demonstrate my skills of full-stack Javascript Web Development.

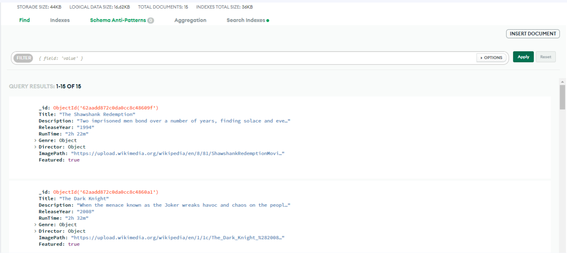
Role: Full-Stack Web Developer

Project Scale: 2.5 Weeks.

Primary Stakeholder: CareerFoundry Web Development Course

Tools Used: MongoDB, Express ,React, Node.Js,Redux, React Bootstrap, Postman, GitHub, Heroku, VS Code.

Process:
To build the app I started with creating the MongoDB database to hold all of the data about the movies, directors and genres.
Then I continued to create the server using Node.js and Express while hosting it on Heroku.



I designed the endpoints for the API for the different CRUD methods and created clear documentation to allow easy use of the API.
Once I tested and verified the endpoints using Postman I was ready to start building the frontend part of the application.

```
/**
 * GET: Returns a list of ALL movies to the user
 * @function [path]/movies/
 * @returns {Object[]} movies
 * @requires passport
 */

// READ:get- Get all movies
app.get('/movies', passport.authenticate('jwt', { session: false }), (req, res) => {
  Movies.find()
    .then((movies) => {
      res.status(200).json(movies);
    })
    .catch((err) => {
      console.error(err);
      res.status(500).send('Error: ' + err);
    });
});
```

I started by creating the necessary components for the UI using React, I then continued to populate the components with data from the API using API calls. After the UI held all of the required information I went ahead and started implementing React Bootstrap to create a consistent and appealing layout.

```
  axios
    .get('https://appformovies.herokuapp.com/movies', {
      headers: { Authorization: `Bearer ${token}` },
    })
    .then((response) => {
      // #4
      this.props.setMovies(response.data);
    })
    .catch(function (error) {
      console.log(error);
    });
  }

  // #4

  render() {
    // #1 movies is extracted from this.props rather than from the this.state
    let { movies, user } = this.props;

    return (
      <Router>
        <Menubar user={user} />
        <Container>
          <Row className="main-view justify-content-md-center">
            <Route
              exact
              path="/"
              render={() => {
                if (!user)
                  return (
                    <Col>
                      <LoginView onLoggedIn={(user) => this.onLoggedIn(user)} />
                    </Col>
```

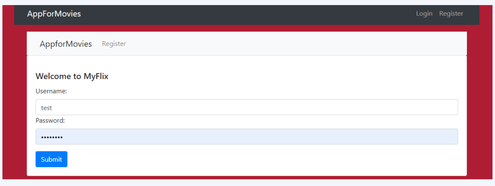
Then I continued to improve the UI after testing the application and optimized the app by implementing Redux to better manage the state of the application. Once all of the functionality and design was implemented I hosted the app live on Netlify.

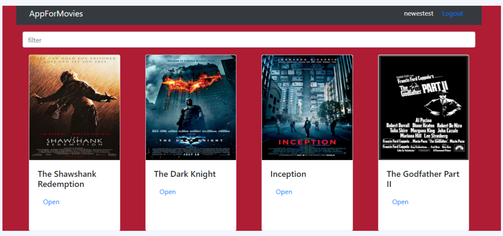
```
export const SET_MOVIES = 'SET_MOVIES';
export const SET_FILTER = 'SET_FILTER';
export const SET_USER = 'SET_USER';

export function setMovies(value) {
  console.log('SET_MOVIES reducer reached');
  return ( type: SET_MOVIES, value );
}

export function setFilter(value) {
  console.log('SET_FILTER reducer reached');
  return ( type: SET_FILTER, value );
}

export function setUser(value) {
  console.log('SET_USER reducer reached');
  return ( type: SET_USER, value );
}
```





Challenges:
This application was both challenging and rewarding, building this app helped me understand how to approach the process of creating a full-stack application from scratch. I've learned how to solve problems by myself through using the knowledge in online developers communities.
I had some challenges with debugging at first since the errors in the console aren't always as clear to understand compared to when developing using Vanilla Javascript but after some practice and tips from my Mentor, I learned how to conquer them one by one.

Project Reflection:
Looking back at the project, I can see that the biggest challenge I was facing was solving errors and bugs by myself, which is an important skill that every web developer should have.
Through this project I had to face that challenge repeatedly and it gave me the confidence to not give up and find the solution by myself.

Another challenge I was facing was creating an appealing and simple design for the app, I went through multiple stages of revisions and changes which helped me prepare for any future challenge I will face.

GitHub Backend

GitHub Frontend

Live Application