

Full Stack Development 2 - Assignment.

Name: David Roche

Features.

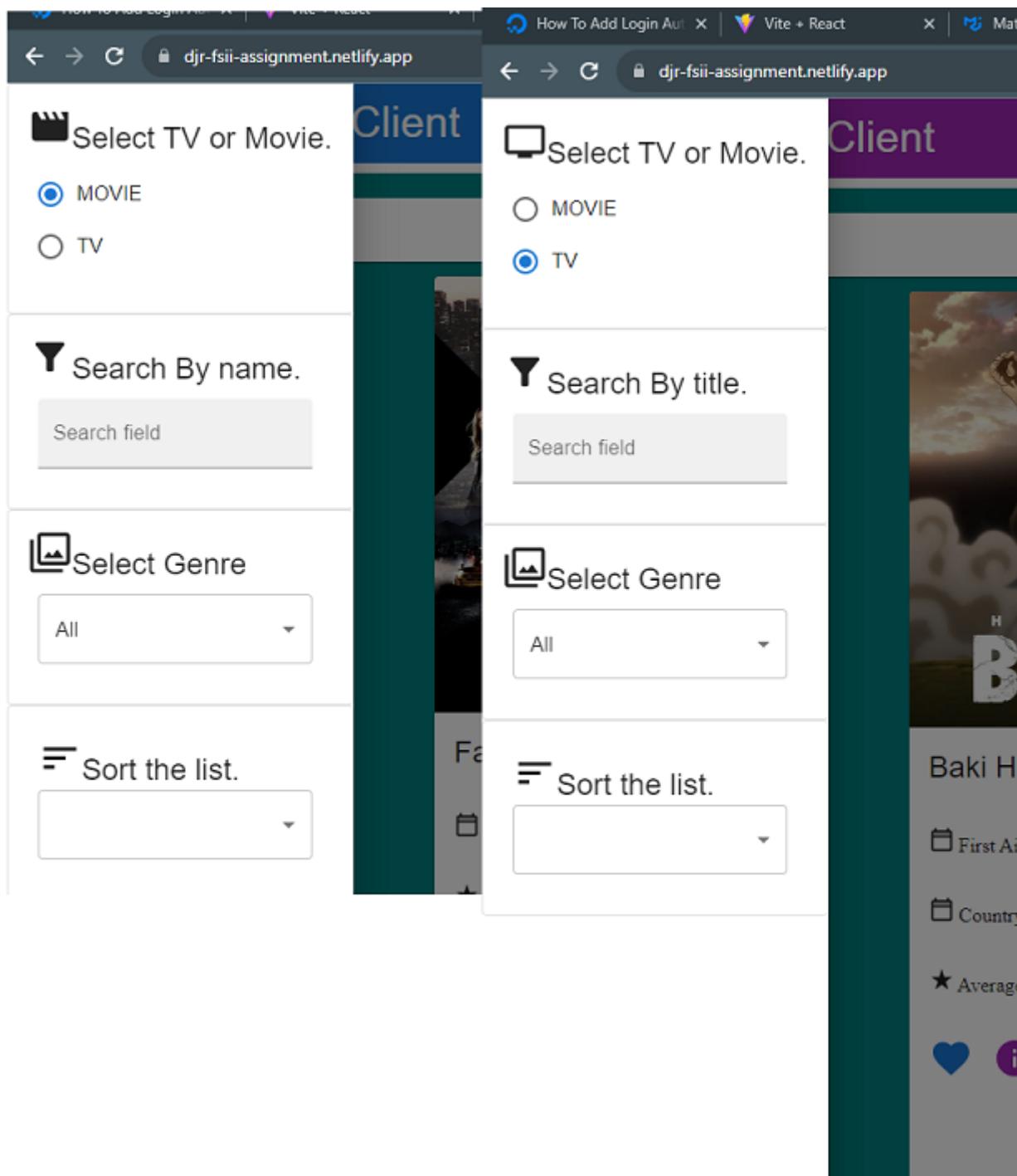
As well as refining some of the features from the class the following new and extended features were added.

- Select between movie or TV shows.
- Revised site header depending on whether Movie or tv selected.
- Revised filter options depending on whether Movie or tv selected.
- Order of the shown movie /TV card selectable via filter.
- TV and movie details page shows available video content for selected movie or TV show.
- Revised TV/ Movie card.

Feature Design.

Select between movie or TV shows..

Allows the user via the filter dialogue to choose between TV or Movie. and stores the option in session information.

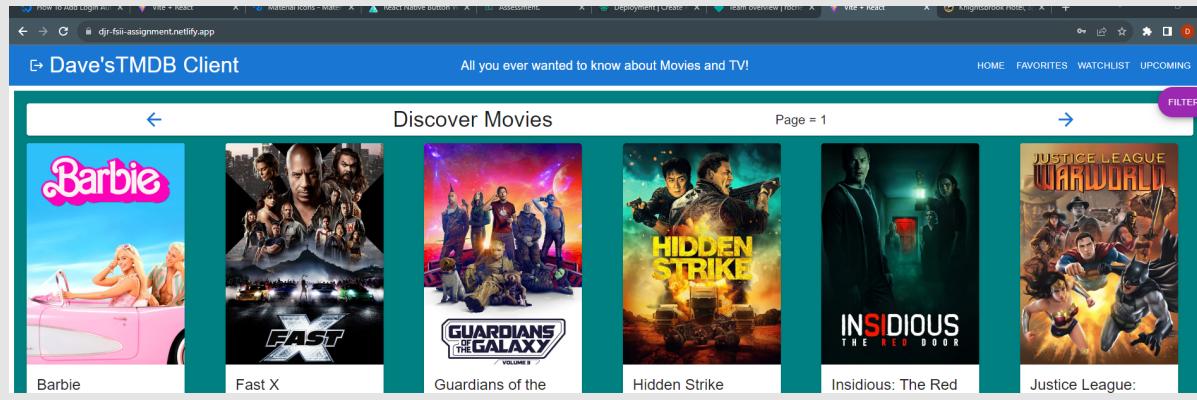


Revised site header depending on whether Movie or tv selected.

Changes the colour and the contents of the site header depending on whether TV or Movie are selected.

The image shows a screenshot of a web browser displaying a movie and TV client interface. The header is purple and contains the text 'Tv Selected' and the title 'E+ Dave'sTMDB Client'. Below the header is a navigation bar with links for 'HOME', 'FAVORITES', 'WATCHLIST', and 'TRENDING'. The main content area features a 'Discover TV' section with several movie and TV show posters, including 'BAKI', 'HANNA BAKI', 'FATAL SEDUCTION', 'Heartstopper', 'King of the Land', and 'Lucifer'. A 'FILTER' button is located in the top right corner of this section. The overall layout is clean and modern, with a focus on readability and user interaction.

Movie Selected



Revised filter options depending on whether Movie or tv selected.

Changes the genre list depending on tv or movie selection.

 Select TV or Movie.

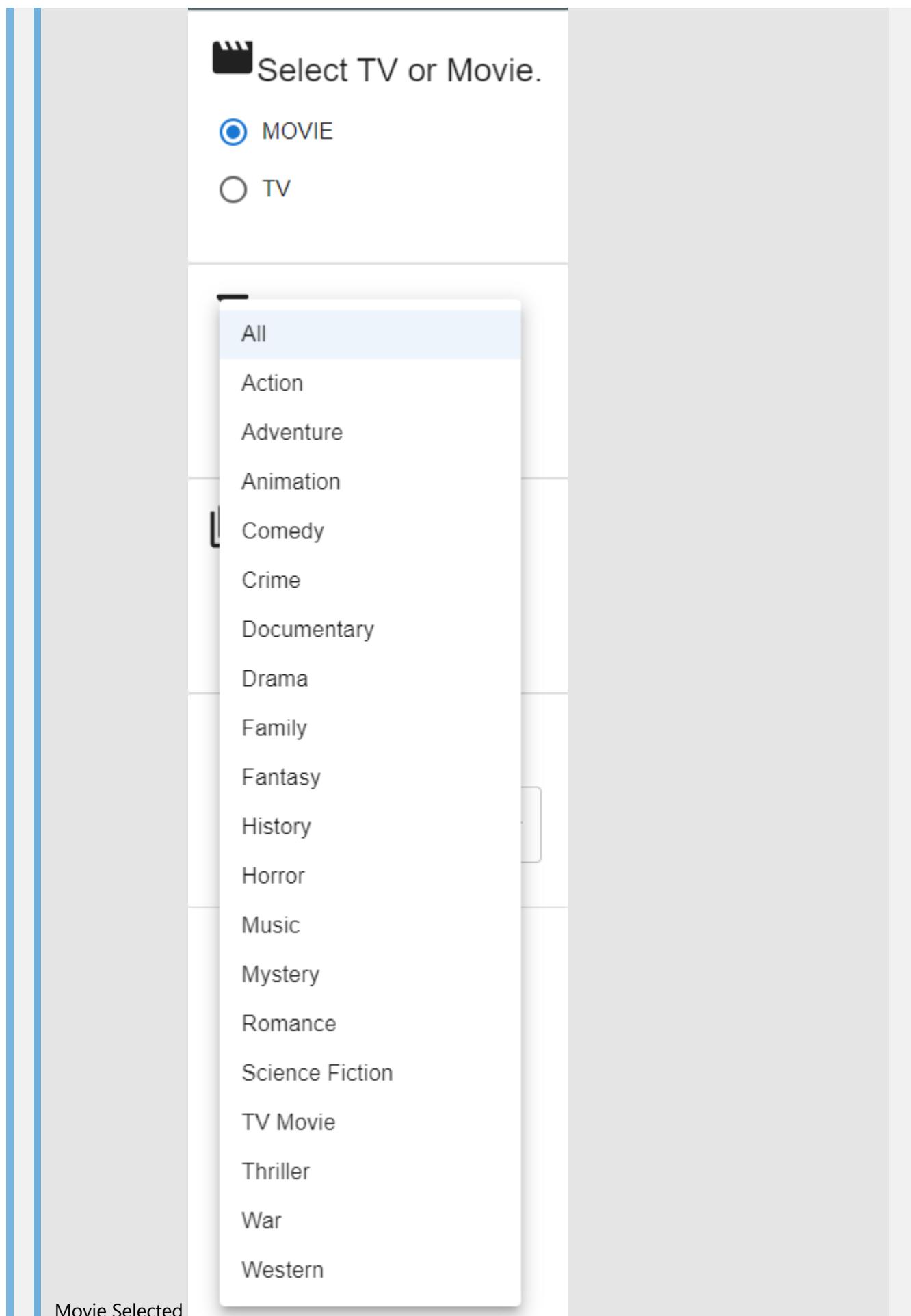
MOVIE
 TV

 Search By title.

Search field

- All
- Action & Adventure
- Animation
- Comedy
- Crime
- Documentary
- Drama
- Family
- Kids
- Mystery
- News
- Reality
- Sci-Fi & Fantasy
- Soap
- Talk
- War & Politics
- Western

Tv Selected



Order of the shown movie /TV card selectable via filter..

Changes the displayed order of TV or Movie cards shown

Movie selected, sorted by release date.

TV selected, sorted by first Air Date

TV and movie details page shows available video content for selected movie or TV show.

User can view movie trailers and clips from the details page

User can filter available videos using the drop down list

Storyboard.

Revised TV and movies card layout and control icons for add to favorites, info and add to watchlist.

Storybook TV card

localhost:6006/?path=/story/home-page-tvcard--basic

Storybook

Find components /

App Header

HOME PAGE

- FilterMoviesCard
 - Default
- MovieCard
 - Default
 - exception
- MovieList
 - Default
- MoviePageHeader
 - Default
- TvCard
 - Default**
 - exception

MOVIE DETAILS PAGE

- MovieDetails
- MovieHeader

 THE WITCHER

The Witcher

First Aired 2019-12-20

Country US

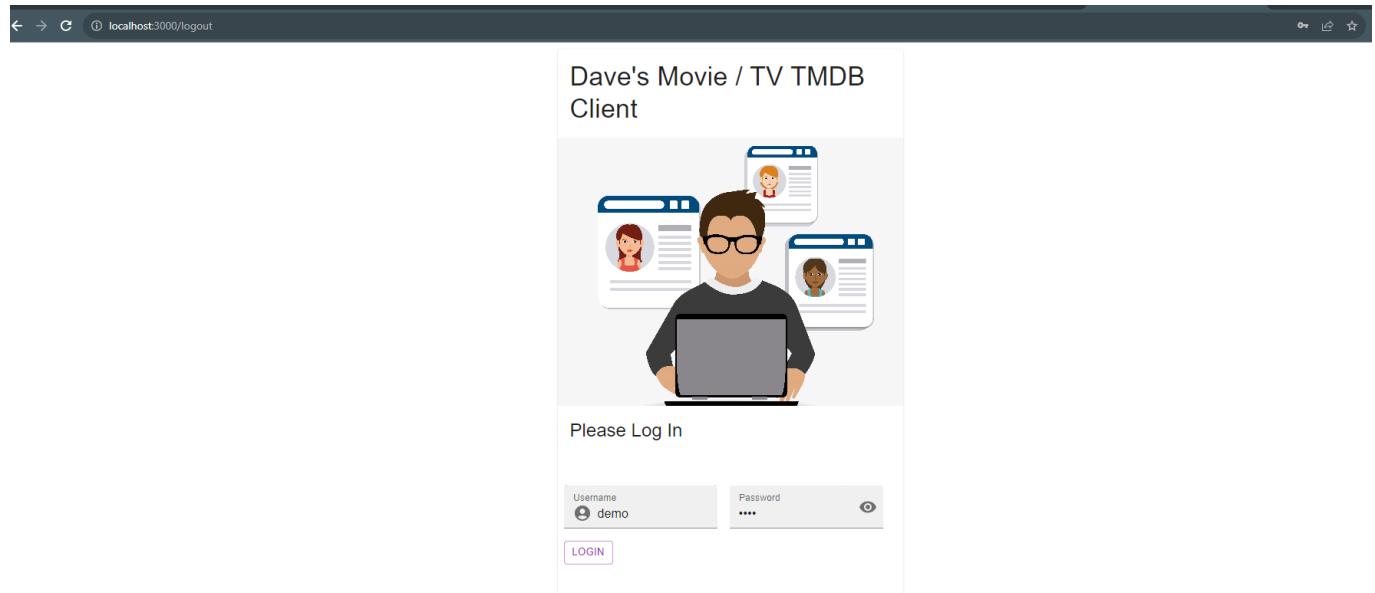
Average Rating 9.155 [Add To Watch list](#)

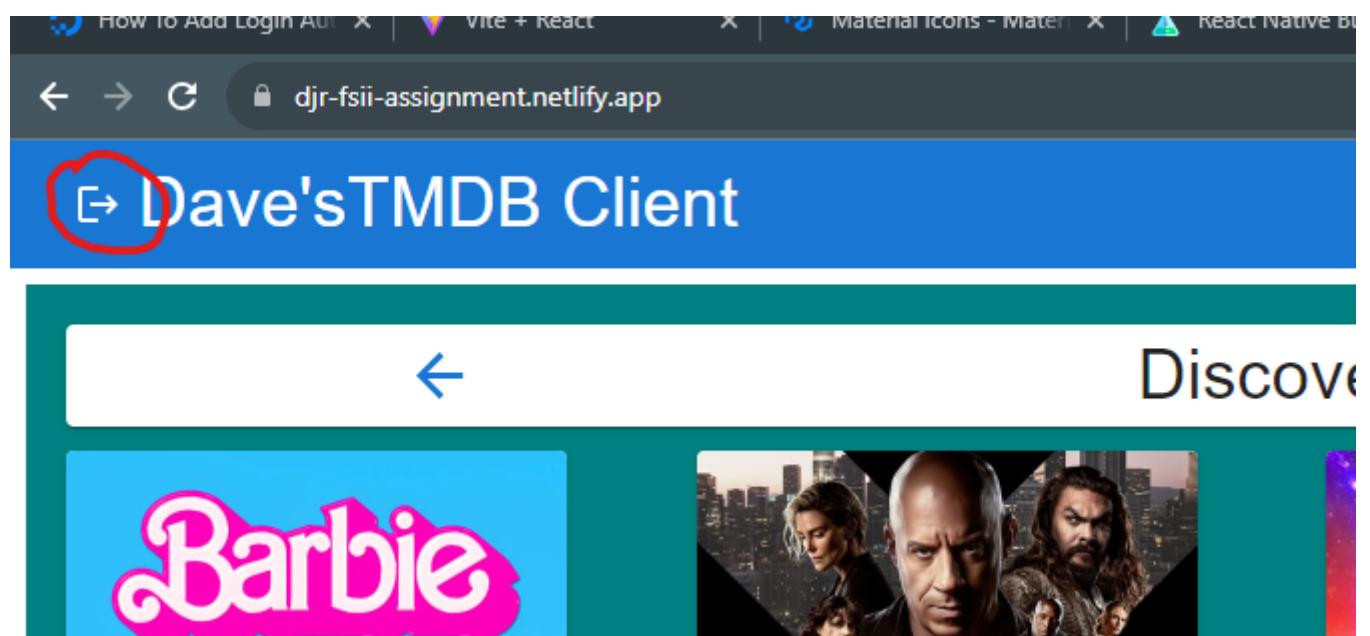
Controls Actions Interactions

Authentication.

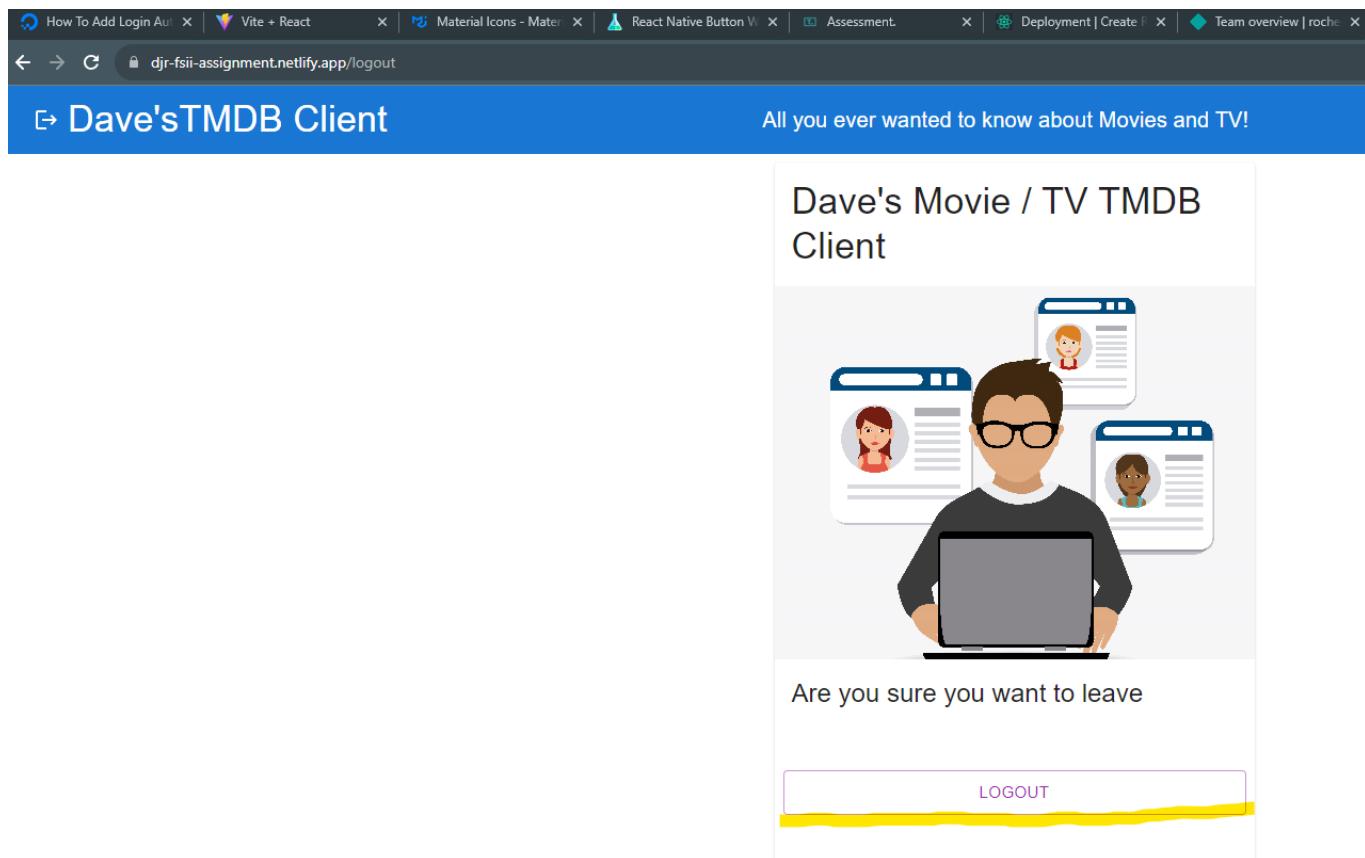
Login Screen



Logout at any stage by clicking this icon



Confirm logout by clicking the logout button here



Protected routes

All routes are protected as the user must login to the site.

Protected functionality.

One logged into the app all options are available. Login is by means of a token stored in sessionStorage.

Reference [./src/components/app/useToken.js](#)

```
import { useState } from "react";

export default function useToken() {
  const getToken = () => {
    const tokenString = sessionStorage.getItem("token");
    const userToken = JSON.parse(tokenString);
    return userToken?.token;
  };

  const [token, setToken] = useState();

  const saveToken = userToken => {
    sessionStorage.setItem('token', JSON.stringify(userToken));
    setToken(userToken.token);
  };

  return {
    setToken: saveToken,
    token
  };
}
```

```
    }  
}
```

Supabase

Not used

Deployment .

The application itself has been deployed to

<https://djr-fsii-assignment.netlify.app/>.

Username: demo ; Password: demo

At the moment the validation middle ware has not been deployed.

Persistence.

No persistance used in this application.

Additional Information.

As well as google search and class notes the following additional resources were used.

[Digital Ocean - Login](#)

[medium.com-JWT aut](#)

[Create React App - Deployment](#)

[Smart Dev - Adding Video](#)

[MUI - Getting started](#)

[MUI - Side Drawer](#)

Middle ware

A middleware app was constructed to allow for user logon and registration.

This app is a follows and ran in node express.

```
const express = require("express");  
const cors = require("cors");  
const mysql = require("mysql2");  
const app = express();  
const bodyParser = require("body-parser");  
  
const db = mysql.createConnection({  
  user: "react",
```

```
host: "localhost",
password: "react",
database: "loginsystem",
});

app.use(cors(),
express.json(),
bodyParser.urlencoded({ extended: true }));

app.post("/register", (req, res) => {
  const username = req.body.username;
  const password = req.body.password;
  db.execute(
    "INSERT INTO users (username, password) VALUES (?,?)",
    [username, password],
    (err, result) => {

      if (err) {
        console.log(err);
        res.send({ err: err });
      }

      if (result.affectedRows == 1) {
        res.send(result);
      } else ({ message: "Error in registering" });
    }
  );
});

app.post("/login", (req, res) => {
  const webReq = req;
  const username = req.body.username;
  const password = req.body.password;

  db.execute(
    "SELECT * FROM users WHERE username = ? AND password = ?",
    [username, password],
    (err, result) => {
      if (err) {
        res.send({ err: err });
      }

      if (result.length > 0) {
        result.message ="log in ok";
        res.send( {
          token: 'Token 1234123',
          id : result.id,
          user : username
        });
      } else
      {
        res.send(result);
      };
    }
})
```

```
 );
});

app.listen(3001, () => {
  console.log("running server");
});
```