



**Bachelor of IT (Computer Science)**  
**Assignment 2 - Client-Side React Application**  
**CAB230 - Web Computing**

*Dane Madsen*  
*n10983864@qut.edu.au*

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose and Description . . . . .	2
1.2	Completeness and Limitations . . . . .	2
<b>2</b>	<b>Use of End Points</b>	<b>2</b>
2.1	/movies/search . . . . .	2
2.2	/movies/data/{imdbID} . . . . .	3
2.3	/people/{id} . . . . .	3
2.4	/user/register . . . . .	3
2.5	/user/login . . . . .	3
2.6	/user/refresh . . . . .	4
2.7	/user/logout . . . . .	4
<b>3</b>	<b>Modules Used</b>	<b>4</b>
3.1	react-router-dom . . . . .	4
3.2	react-responsive-carousel . . . . .	4
3.3	ag-grid-react . . . . .	4
3.4	ag-grid-community . . . . .	4
<b>4</b>	<b>Application Design</b>	<b>5</b>
4.1	Navigation and Layout . . . . .	5
4.2	Usability and Quality of Design . . . . .	5
4.3	Accessibility . . . . .	6
<b>5</b>	<b>Technical Description</b>	<b>6</b>
5.1	Architecture . . . . .	6
5.2	Test Plan . . . . .	8
5.3	Difficulties, Exclusions or Unresolved and Persistent Errors . . . . .	8
<b>6</b>	<b>User Guide</b>	<b>8</b>
<b>7</b>	<b>Appendix</b>	<b>9</b>
7.1	Appendix A - Introduction . . . . .	9
7.2	Appendix B - Use of End Points . . . . .	10
7.3	Appendix C - Application Design . . . . .	13
7.4	Appendix D - Test Plan . . . . .	15

# 1 Introduction

## Appendix A

### 1.1 Purpose and Description

The purpose of this react application is to collate and display information regarding movies, and cast members to the user in a responsive and accessible manner. The application should allow the user to search for movies by year or title and display the results in a list (Figure 1). When a movie is selected, the application should display all the details of the movie, including the year the movie was made, the plot of the movie and all the cast members involved with the movie (Figure 2). The application should also allow the user to visit individual pages of cast members where the user can view that cast members details, including their birth year, death year and all the movies they have been involved with.

It achieve this goal and to provide the user with the best experience I can, i have used a number of advanced react features, including the use of react router, IntesectionObserver and react-responsive-carousel. The use of these features has allowed me to create a professional looking application that is both easy to use and responsive for the user.

### 1.2 Completeness and Limitations

This implementation covers all the requirements of the assignment specification. Navigation is handled using react router, controlled forms are used for all user input, and every 10 minutes the refresh token is used to get a new access token. In addition to this, the application uses ag-grid to display the search results in an infinitely scrolling grid and react-responsive-carousel to display the highest scoring movies of all time on the home (landing) page. On the person page the movies the person is involved in are displayed in a grid with the IMDb Rating.

# 2 Use of End Points

## Appendix B

The functionality for all the API endpoints is handled in API.js. This file contains all the functions that are used to make requests to the API. By containing all the API functionality in one file, it makes it easier to maintain and potentially update the application in the future.

### 2.1 /movies/search

This endpoint is implemented as the getMovies function and is utilised by two search forms at the top of the application. The first search form is for the user to search for movies by title, and the second is for the user to search for movies by year. Both of these

forms are controlled forms and can be accessed from any page in the application. Doing it this way removes an extra layer of complexity for the user and allows them to get into the function of the application right from the start of the application.  
(Figure 3)

## **2.2 /movies/data/{imdbID}**

This endpoint is implemented as the `getMovie` function and is utilised by the movie page to get the details of the movie the user has selected. The movie page is accessed by clicking on any of the movies in the search results. The movie page displays all the details of the movie, including the title, year, box office earnings, runtime, genre, country of origin, plot, cast and ratings. each cast member is a clickable link that will lead to that cast members respective page.  
(Figure 4)

## **2.3 /people/{id}**

This endpoint is implemented as the `getPerson` function and is utilised by the person page to get the details of the person the user has selected. The person page is accessed by clicking on any of the cast members on the movie page. The person page displays all the details of the person exposed by the API, including their name, birth year, death year, all the movies they have been involved with. In a grid the name of each movie, the name of the character the person played and the IMDb Rating is displayed for the users convenience.  
(Figure 5)

## **2.4 /user/register**

This endpoint is implemented as the `postRegister` function and is utilised by the profile page to register new users. The profile page is accessed by clicking on the profile button in the navigation bar at the top of the page. When first launched, if the user isnt logged in the profile page will be in the login state, if the user presses the register button the profile page will be transitioned into the register state. The register state contains fields for the user to enter their email, password and confirm password. If the user presses the register button the `/user/register` endpoint will be utilised to register the user and the tokens will be stored in local storage. If the user presses back to login the page will be transitioned back into the login state.  
(Figure 6)

## **2.5 /user/login**

This endpoint is implemented as the `postLogin` function and is utilised by the profile page to login existing users. If the user accesses the the profile page while logged out they will be greeted with a login prompt with an email and password field, button to login and a button to register. If the user presses the register button the page will be transitioned into the register state. If the user presses the login button the `/user/login` endpoint will be utilised to login the user and the tokens will be stored in local storage.  
(Figure 7)

## **2.6 /user/refresh**

This endpoint is implemented as the postRefresh function and is utilised in the back end in App.js to refresh the tokens when the app first starts and every 10 minutes after. This is done to ensure the user does not have to login every 10 minutes.

## **2.7 /user/logout**

This endpoint is implemented as the postLogout function and is utilised by the profile page to logout the user. If the user accesses the the profile page while logged in they will be greeted with a message telling them they are logged in accompanied with a button to logout, if the user presses the logout button the /user/logout endpoint will be utilised to logout the user and the tokens will be deleted.

(Figure 8)

# **3 Modules Used**

## **3.1 react-router-dom**

This module was used to handle navigation within the application. It was used to create the navigation bar at the top of the page and to create the routes for the different pages in the application.

<https://www.npmjs.com/package/react-router-dom>

## **3.2 react-responsive-carousel**

This module was used to create a carousel on the home page to display the movies with the highest IMDb Rating.

<https://www.npmjs.com/package/react-responsive-carousel>

## **3.3 ag-grid-react**

This module was used in MoviesPage to create a grid to display the search results in an infinitely scrolling grid, and in PersonPage it is used to display the movies the person stars in.

<https://www.npmjs.com/package/ag-grid-react>

## **3.4 ag-grid-community**

This module was used in both MoviesPage and PersonPage to utilise the CSS for the respective grids.

<https://www.npmjs.com/package/ag-grid-community>

## 4 Application Design

### Appendix C

#### 4.1 Navigation and Layout

The application has been designed to be as simple to use as possible. The ultimate goal of the design was to develop a user experience that the user can understand and use without having to think about it.

A few design choices were made to achieve this goal. The first major choice I made was to move the search bars to the navigation bar at the top of the page. This was done to make the search functionality more accessible to the user by allowing them to search from any page in the application. Moving the search bars to the navigation also freed up space on the MoviesPage allowing me to display more search results at once.

During development I also investigated the possibility of having the poster image for each movie displayed in the search results (Figure 10, Figure 11). I decided against this as it excessively queries the API and often resulted in a rate limit. It also increased the complexity of being able to sort the search results which was something I wanted to implement.

#### 4.2 Usability and Quality of Design

As the application was designed with simplicity in mind, the usability of the application is very high. The application is very easy to use and the user can navigate the application with ease. The application is laid out in a way that is very clean, the only page that can at all be considered cluttered is the LandingPage because of the carousel but even then the carousel is very clean, easy to use and it doesn't seem to distract the user from the rest of the page.

The navigation of the app is very simple and easy for the user to understand. As previously mentioned, by having the search bars within the navigation bar at the top of the page the user can search from anywhere in the application. The only part of the navigation that may confuse the user is the home button using the logo of the application. It may not be clear to the user that this is a button that leads them home. But, this is quite a common design choice in web design and there isn't much use for the user to return to the home page as the user can search from any page in the application.

The design of the application is consistent throughout. The navigation bar is the same on every page, very little colours are used throughout the application and the colours that are used are consistent. Only the standard font is used throughout the application except in the ag-grids where a separate style is used.

## 4.3 Accessibility

The application has attempted to be as accessible as possible. The application achieve most of the checkpoints of the W3C Web Content Accessibility Guidelines 2.0. The application has achived the following checkpoints:

- 1.1.1 All non-text content such as images have alternative text.
- 1.2.1 Information conveyed in colour can also be conveyed in black and white.
- 1.6.1 Page can be read without CSS.
- 1.6.2 Text equivalents update when content changes.
- 1.7.1 The screen doesnt flicker.
- 1.14.1 The application uses the clearest language appropriate.

## 5 Technical Description

### 5.1 Architecture

- **Components:**  
This folder contains the components and the API handler for the application.
- **Pages:**  
This folder contains all the pages for the application.
- **Styles:**  
This folder contains all the CSS files for the application.

The diagram on the next page show the file structure of the application. As you can see, the application is structured in a very simple way. As previously mentioned, all the code for handling API requests is contained within the API.js file. This has made it easier to write the code for the application as it is all in one place and can be reused easily.

The only component of the application that has been seperated out into its own file is the search inputs. The only reason this was done was because I couldnt get it working within App.js.

Each page of the application has its own JS and CSS file. This was done to make the code easier to read and understand, and to cut down on the amount of code in each file.

Project Root

- Docs
- Public
- SRC
- .gitignore
- package-lock.json
- package.json
- README.md

SRC

- Components
  - SearchInputs.js
  - API.js
- Pages
  - LandingPage.js
  - MoviePage.js
  - MoviesPage.js
  - PersonPage.js
  - ProfilePage.js
- Styles
  - App.css
  - LandingPage.css
  - MoviePage.css
  - MoviesPage.css
  - PersonPage.css
  - ProfilePage.css
  - SearchInputs.css
- App.js
- Logo.svg



## 5.2 Test Plan

### Appendix D

Task	Expected Outcome	Result	Screenshot
Search by title.	List of movies returned	PASS	Figure 13
Search by year.	List of movies returned	PASS	Figure 14
Search by title and year.	Movie, or List of movies returned	PASS	Figure 15
Click on a movie.	Respective movie page is displayed.	PASS	Figure 16
Click on a cast member. (Logged Out)	User is prompted to Login	PASS	Figure 17
Click on a cast member. (Logged In)	Respective person page is displayed.	PASS	Figure 18
Register a new user.	User is registered and logged in.	PASS	Figure 19 Figure 20
Login as a user.	User is logged in.	PASS	Figure 21 Figure 22
Log out as a user.	User is logged out.	PASS	Figure 23 Figure 24

### 5.3 Difficulties, Exclusions or Unresolved and Persistent Errors

The only notable issues that persist in the final version of the app is the lack of an ability to sort the movies by title, year or rating, and an error that occurs when the application has been offline for more than a day. The first issue is due to the fact that the API doesn't support sorting and the second issue is due to the fact that the refresh token expires after 24 hours. In summation, both of these issues are server side so there's not a lot that can be done.

## 6 User Guide

## 7 Appendix

### 7.1 Appendix A - Introduction

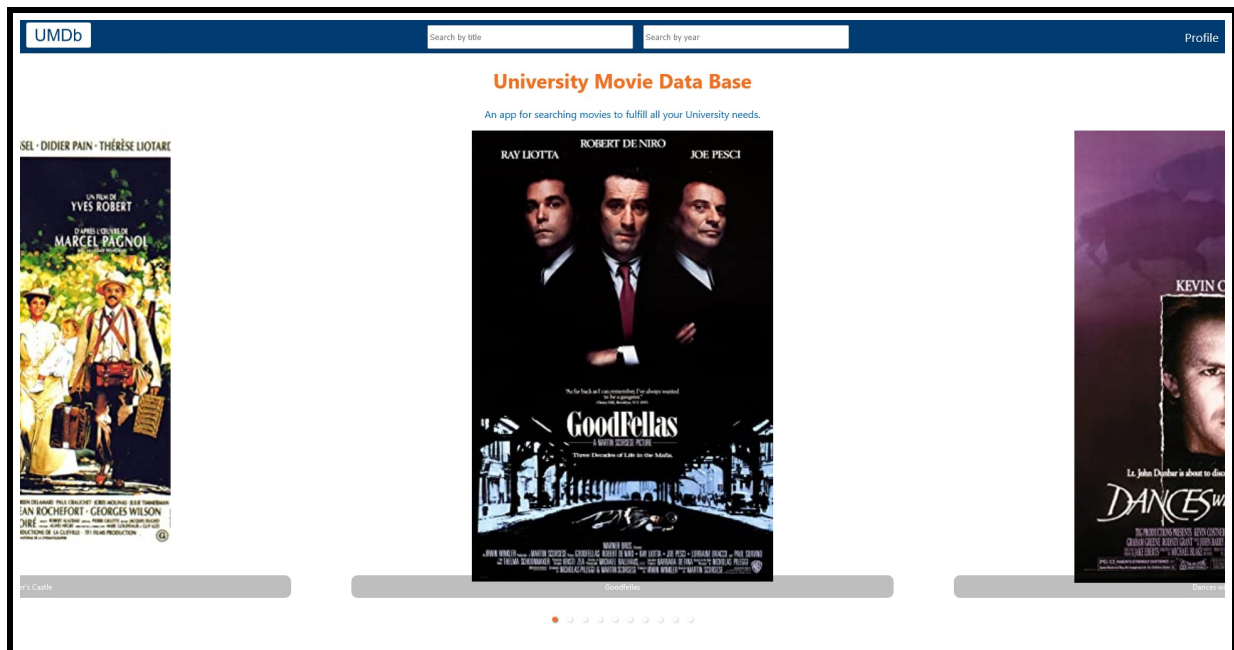


Figure 1: Landing Page of the Application

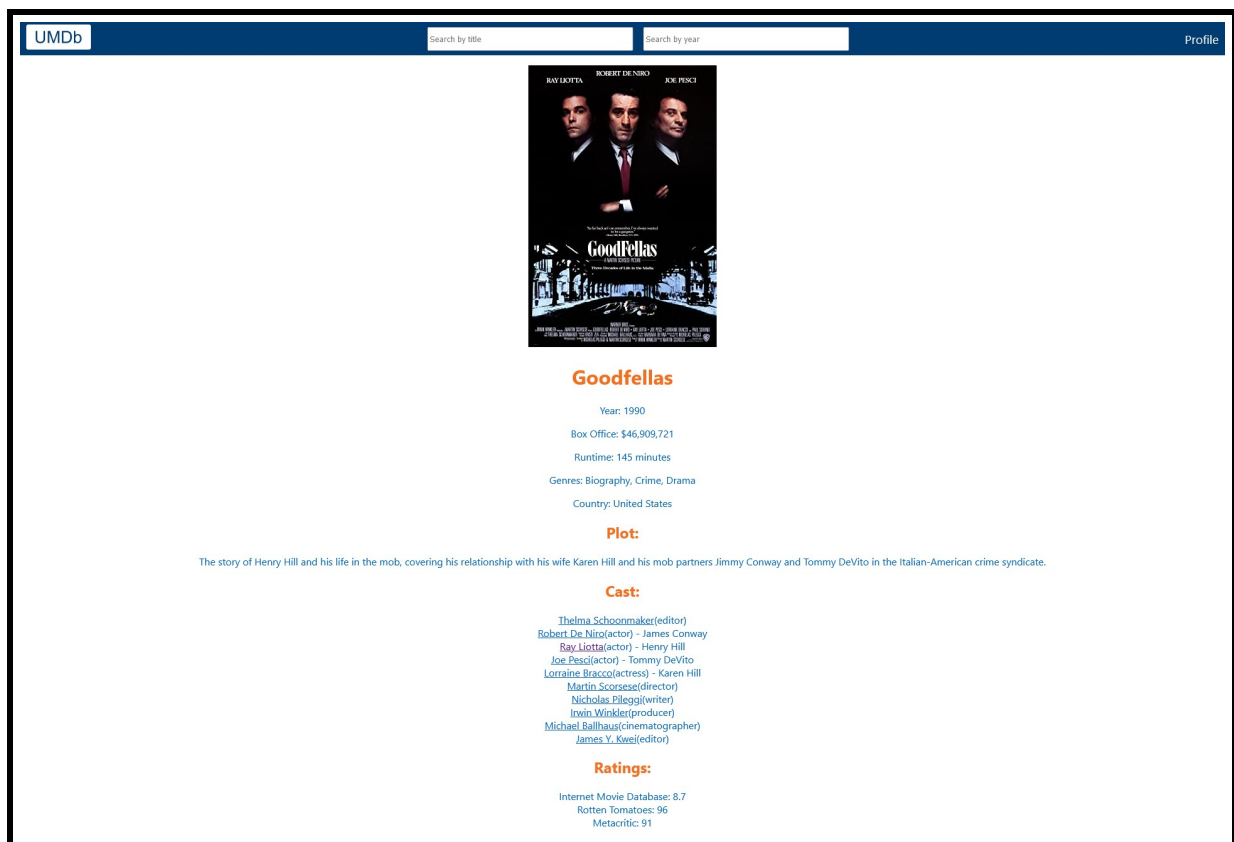


Figure 2: Movie Page of the Movie "Goodfellas"

UMDb			Search by title	Search by year	Profile
Person Details					
Kevin Costner					
Date of Birth: 1955					
Death:					
Roles:					
Movie Name	Character	IMDb Rating			
Dances with Wolves	Lieutenant Dunbar	8			
Revenge	Cochran	6.2			
JFK	Jim Garrison	8			
Robin Hood: Prince of Thieves	Robin of Locksley	6.9			
The Bodyguard	Frank Farmer	6.3			
A Perfect World	Butch Haynes	7.5			
Rapa Nui		6.4			
The War	Stephen Simmons	6.7			

Figure 3: Person Page of the Actor "Kevin Costner"

## 7.2 Appendix B - Use of End Points

UMDb						Search by title	Search by year	Profile
Title	Year	IMDb Rating	Rotten Tomatoes Rating	Metacritic Rating	Classification			
Goodfellas	1990	8.7	96	91	R			
A Few Good Men	1992	7.7	84	62	R			
The Good Son	1993	6.4	25	45	R			
The Long Kiss Goodnight	1996	6.8	68	44	R			
Good Burger	1997	5.7	33	41	PG			
Good Will Hunting	1997	8.3	97	70	R			
Midnight in the Garden of Good and Evil	1997	6.6	50	57	R			
As Good as It Gets	1997	7.7	86	67	PG-13			
Good Advice	2001	6.2	33		R			
The Good Girl	2002	6.4		71	R			
The Good Thief	2002	6.5	77	68	R			
No Good Deed	2002	5.4	25		R			
Good Bye Lenin!	2003	7.7	90	68	R			
The Good Shepherd	2006	6.7	56	61	R			
Goodbye, Dragon Inn	2003	7.1	81	83	N/A			
A Good Woman	2004	6.5	37	53	PG			
In Good Company	2004	6.5	82	66	PG-13			
A Good Year	2006	6.9	26	47	PG-13			

Figure 4: Movies Page of the Application

UMDb

Fight C

Search by year

Profile

## Fight Club

Year: 1999

Box Office: \$37,030,102

Runtime: 139 minutes

Genres: Drama

Country: Germany, United States

**Plot:**

An insomniac office worker and a devil-may-care soap maker form an underground fight club that evolves into much more.

**Cast:**

[Art Linson](#)(producer)  
[Brad Pitt](#)(actor) - Tyler Durden  
[Edward Norton](#)(actor) - Narrator  
[Meat Loaf](#)(actor) - Robert Paulsen  
[Zach Grenier](#)(actor) - Richard Chesler (Regional Manager)  
[David Fincher](#)(director)  
[Chuck Palahniuk](#)(writer)  
[Jim Uhls](#)(writer)  
[Ross Grayson Bell](#)(producer)  
[Cedric Chaffin](#)(producer)

**Ratings:**

Internet Movie Database: 8.8  
 Rotten Tomatoes: 79  
 Metacritic: 66

Figure 5: Movie Page of the Movie "Fight Club"

UMDb

Search by title

Search by year

Profile

## Person Details

### Robert De Niro

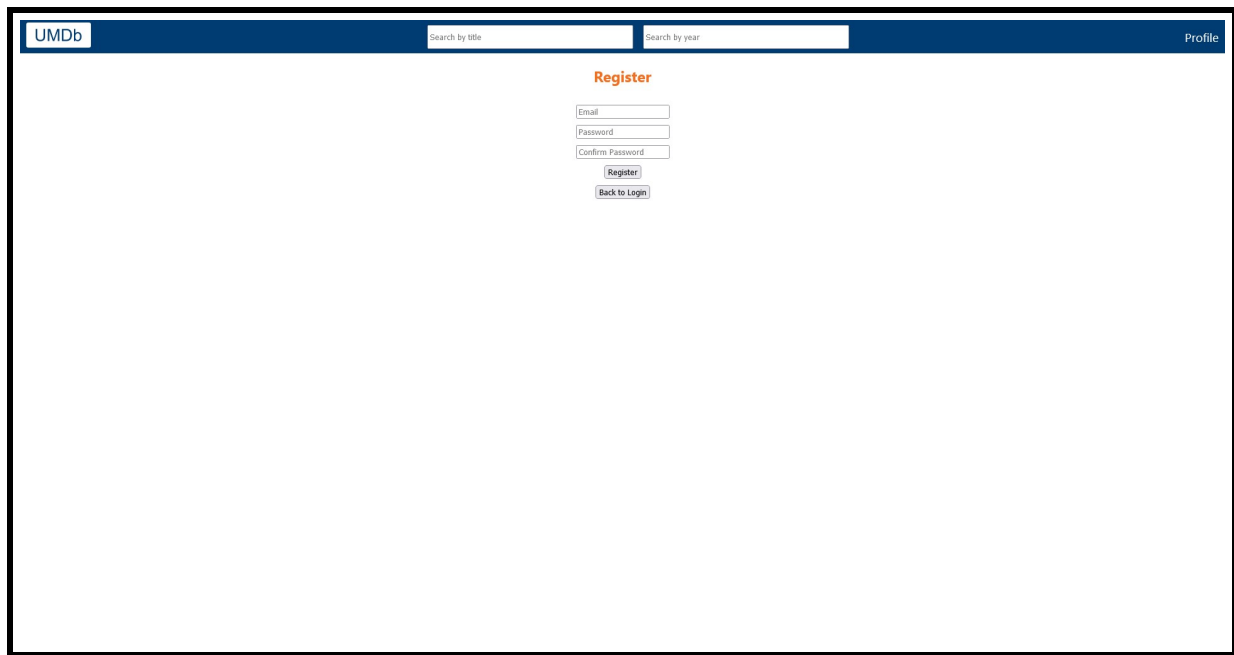
Date of Birth: 1943

Death:

**Roles:**

Movie Name	Character	IMDb Rating
Awakenings	Leonard Lowe	7.8
Goodfellas	James Conway	8.7
Stanley & Iris	Stanley Cox	6.3
Backdraft	Donald Rimgale	6.7
Cape Fear	Max Cady	7.3
Guilt by Suspicion	David Merrill	6.5
Thunderheart		6.8
A Bronx Tale	Lorenzo	7.8

Figure 6: Person Page of the Actor "Robert De Niro"



UMDb

Search by title

Search by year

Profile

**Register**

Email

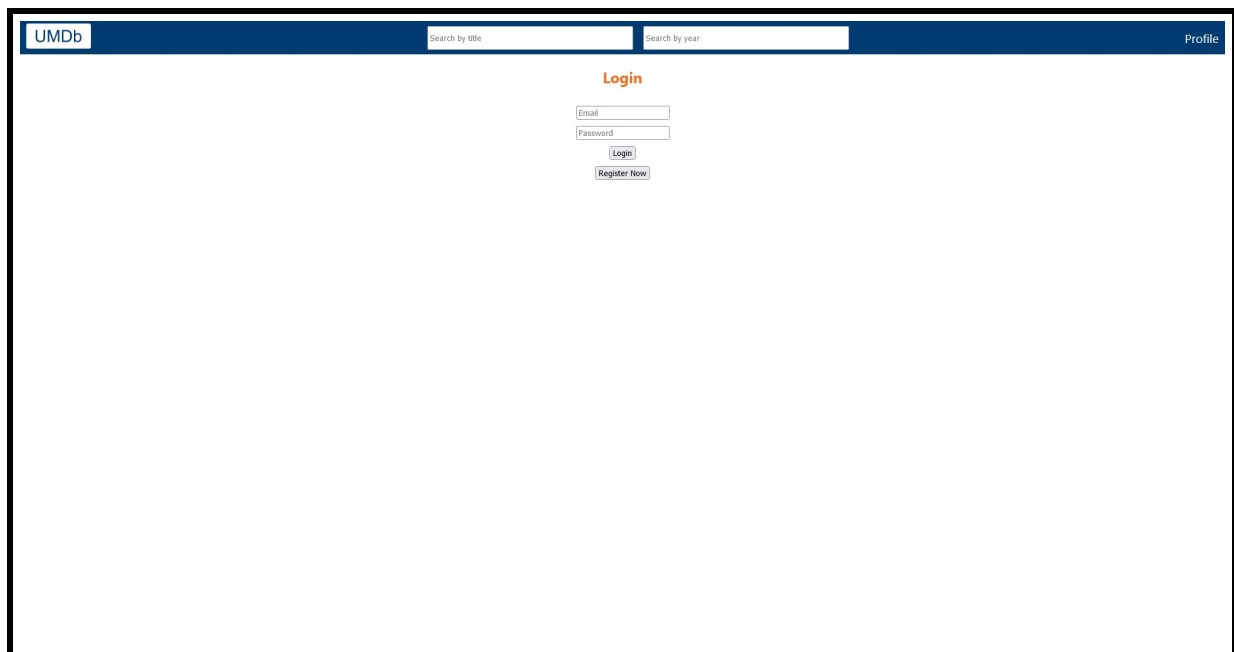
Password

Confirm Password

Register

Back to Login

Figure 7: Profile Page in the Register State



UMDb

Search by title

Search by year

Profile

**Login**

Email

Password

Login

Register Now

Figure 8: Profile Page in the Login State

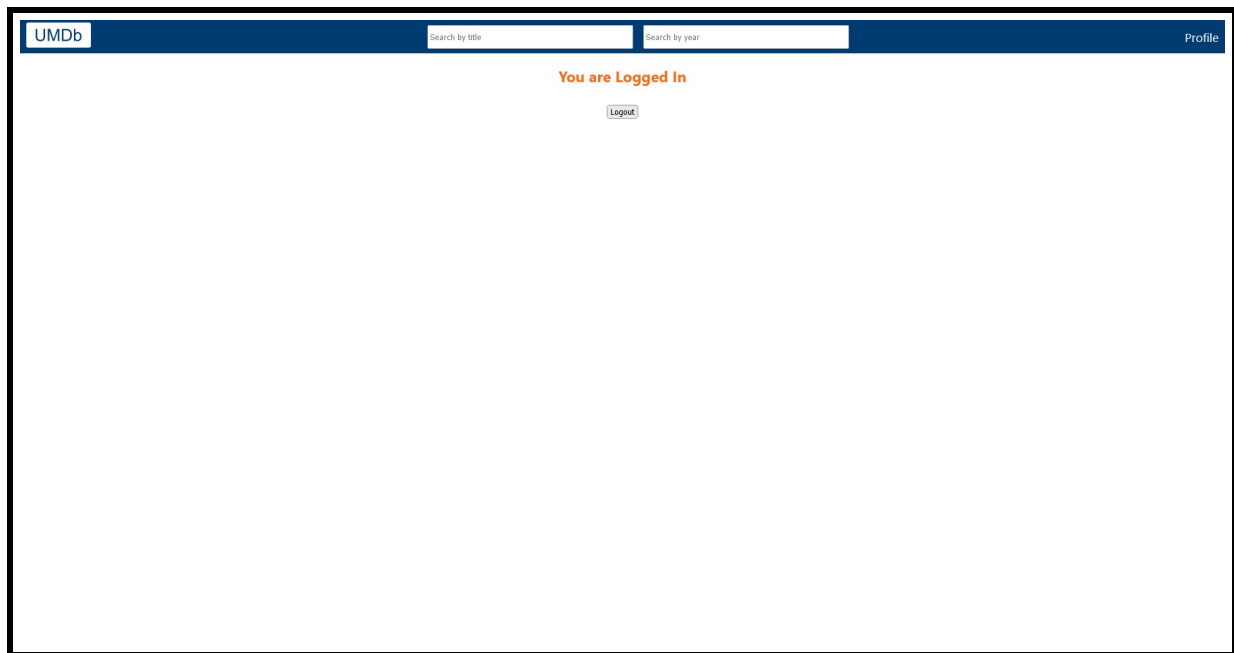


Figure 9: Profile Page Ready for the User to Logout

## 7.3 Appendix C - Application Design

The screenshot shows the UMDb Movies Page. At the top, there is a dark blue header with the UMDb logo on the left, the word "Movies" in the center, and a "Profile" link on the right. Below the header, the word "Movies" is displayed in orange. Below this, there are search input fields for "Search by Title:" and "Search by Year:" with a "Search" button. The "Search by Year:" field contains the value "1996". Below the search fields is a table of movies from 1996.

Title	Year	IMDb Rating	Rotten Tomatoes Rating	Metacritic Rating	Classification
<a href="#">The Pillow Book</a>	1996	6.5	67	64	Not Rated
<a href="#">Poison Ivy II</a>	1996	4.4	14		R
<a href="#">Tromeo and Juliet</a>	1996	6			R
<a href="#">101 Dalmatians</a>	1996	5.7	41	49	G
<a href="#">2 Days in the Valley</a>	1996	6.5	62	56	R
<a href="#">The Adventures of Pinocchio</a>	1996	5.2	35		G
<a href="#">Alaska</a>	1996	5.7	23		PG
<a href="#">Albino Alligator</a>	1996	6.1	50	48	R
<a href="#">All Dogs Go to Heaven 2</a>	1996	5.5	20		G
<a href="#">The Apartment</a>	1996	7.4			R
<a href="#">The Arrival</a>	1996	6.2	66	56	PG-13
<a href="#">The Associate</a>	1996	6.1	28		PG-13
<a href="#">Bad Moon</a>	1996	5.8	30		R
<a href="#">Barb Wire</a>	1996	3.5	28	40	R
<a href="#">Basquiat</a>	1996	6.9		65	R
<a href="#">Beautiful Girls</a>	1996	7.1	78	64	R
<a href="#">Beautiful Thing</a>	1996	7.6	91		R
<a href="#">Beavis and Butt-Head Do America</a>	1996	6.8	71	64	PG-13
<a href="#">Bed of Roses</a>	1996	6	19		PG

Figure 10: An earlier version of the MoviePage

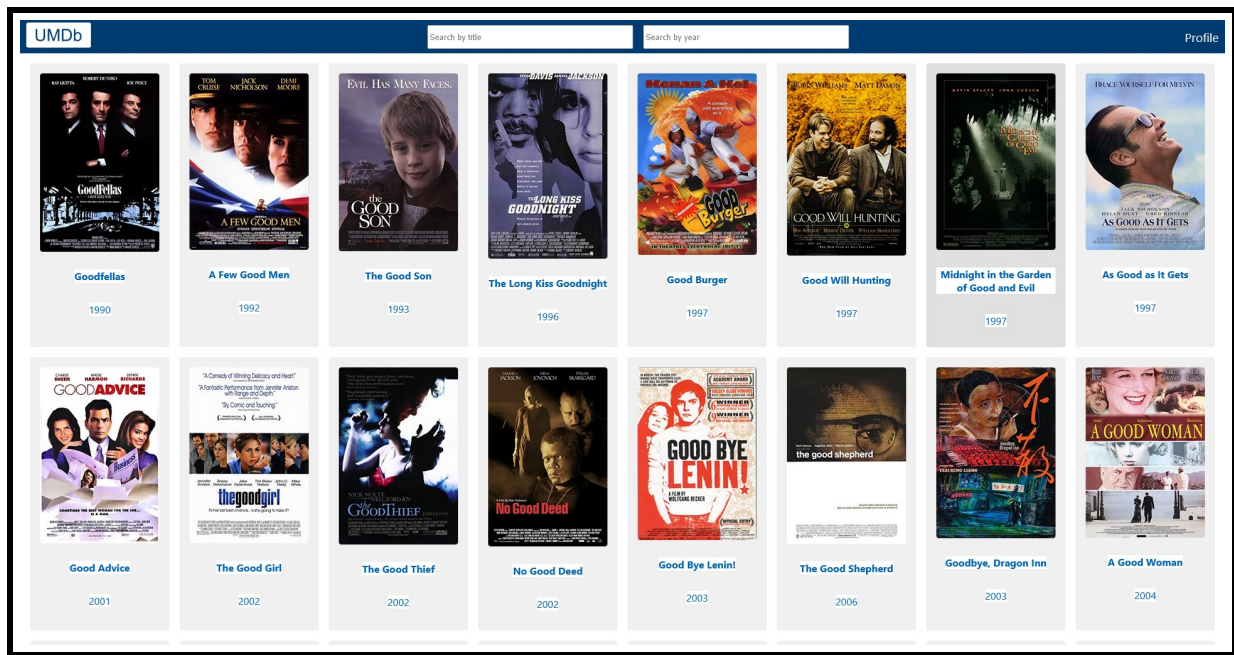


Figure 11: An experimental version of the MoviesPage

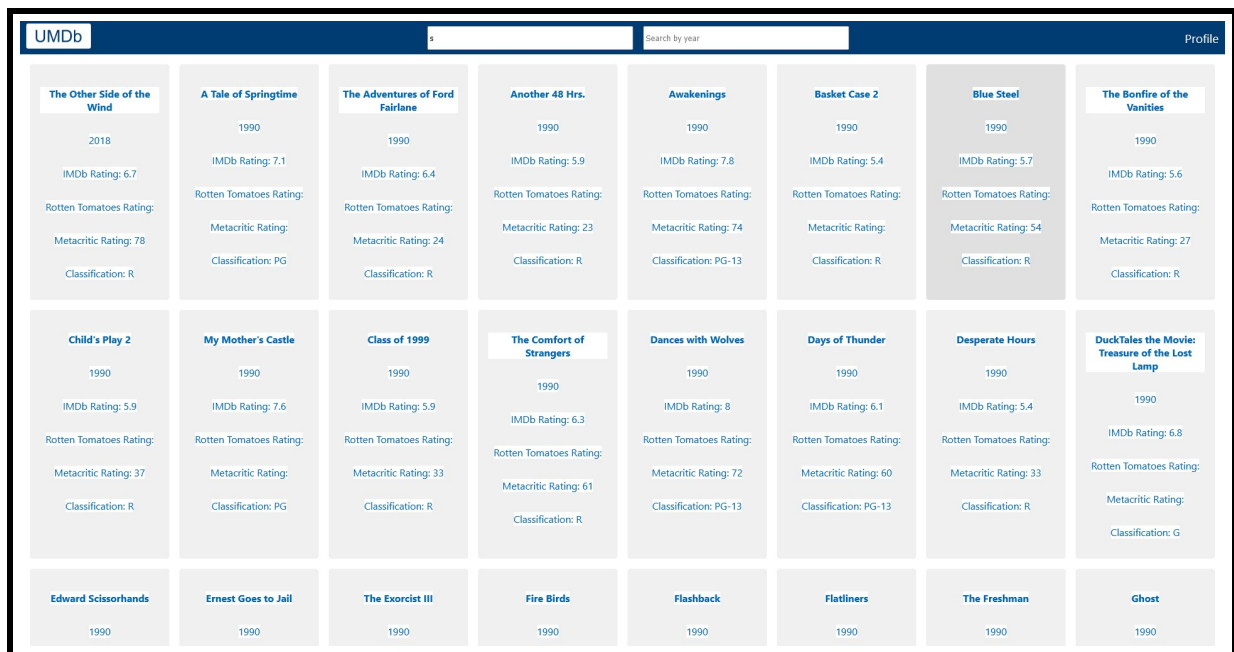


Figure 12: Another experimental version of the MoviesPage

## 7.4 Appendix D - Test Plan

UMDb		Blade	Search by year		Profile	
Title	Year	IMDb Rating	Rotten Tomatoes Rating	Metacritic Rating	Classification	
Sing Blade	1996	8	97	84	R	
Blade	1998	7.1	57	47	R	
Blade II	2002	6.7	57	52	R	
Blade: Trinity	2004	5.8	24	38	R	
The Hidden Blade	2004	7.7	84	76	R	
Blades of Glory	2006	6.3	70	64	PG-13	
Shinobi: Heart Under Blade	2005	6.8			R	
14 Blades	2010	6.3	69	50	R	
Blade Runner 2049	2017	8	88	81	R	
Dragon Blade	2015	5.9	35	41	R	
Blade of the Immortal	2017	6.7	86	72	R	

Figure 13: Test 1 - Search by Title

UMDb		Search by title	2008	Profile	
Title	Year	IMDb Rating	Rotten Tomatoes Rating	Metacritic Rating	Classification
The Bank Job	2008	7.2	80	69	R
Berlin Calling	2008	7.2			N/A
Lymelife	2008	6.8	63	64	R
Babylon A.D.	2008	5.5	7	26	PG-13
The Red Baron	2008	6.3	17		PG-13
Indiana Jones and the Kingdom of the Crystal Skull	2008	6.2	78	65	PG-13
Four Christmases	2008	5.7	25	41	PG-13
Iron Man	2008	7.9	94	79	PG-13
Journey to the Center of the Earth	2008	5.8	61	57	PG
Cher: Part Two	2008	6.8	79	64	Not Rated
Leatherheads	2008	6	52	56	PG-13
Synecdoche, New York	2008	7.5	69	67	R
The Onion Movie	2008	6.2			Not Rated
The Secret of Moonacre	2008	6	23		PG
Bolt	2008	6.8	90	67	PG
Far Cry	2008	3.2			R
My Sassy Girl	2008	6.2			PG-13
Rudo y Cursi	2008	6.7	73	67	R


Figure 14: Test 2 - Search by Year



UMDb		The D	2008	Profile		
Title	Year	IMDb Rating	Rotten Tomatoes Rating	Metacritic Rating	Classification	
The Dark Knight	2008	9	94	84	PG-13	
The Mummy: Tomb of the Dragon Emperor	2008	5.2	13	31	PG-13	
The Duchess	2008	6.9	62	62	PG-13	
I Sell the Dead	2008	6	73	62	Not Rated	
Dance of the Dead	2008	5.8	80		R	
The Day the Earth Stood Still	2008	5.5	20	40	PG-13	

Figure 15: Test 3 - Search by Title and Year

UMDb
Blade
Search by year:
Profile



### Blade Runner 2049

Year: 2017

Box Office: \$92,071,675

Runtime: 164 minutes

Genres: Action, Drama, Mystery

Country: United States, United Kingdom, Canada, Spain

**Plot:**

Young Blade Runner K's discovery of a long-buried secret leads him to track down former Blade Runner Rick Deckard, who's been missing for thirty years.

**Cast:**

[Andrew A. Kosove](#)(producer)  
[Harrison Ford](#)(actor) - Rick Deckard  
[Ryan Gosling](#)(actor) - "K"  
[Ana de Armas](#)(actress) - Joi  
[Dave Bautista](#)(actor) - Sapper Morton  
[Denis Villeneuve](#)(director)  
[Hampton Fancher](#)(writer)  
[Michael Green](#)(writer)  
[Philip K. Dick](#)(writer)  
[Broderick Johnson](#)(producer)

**Ratings:**

Internet Movie Database: 8  
Rotten Tomatoes: 88  
Metacritic: 81

Figure 16: Test 4 - Click on a Movie

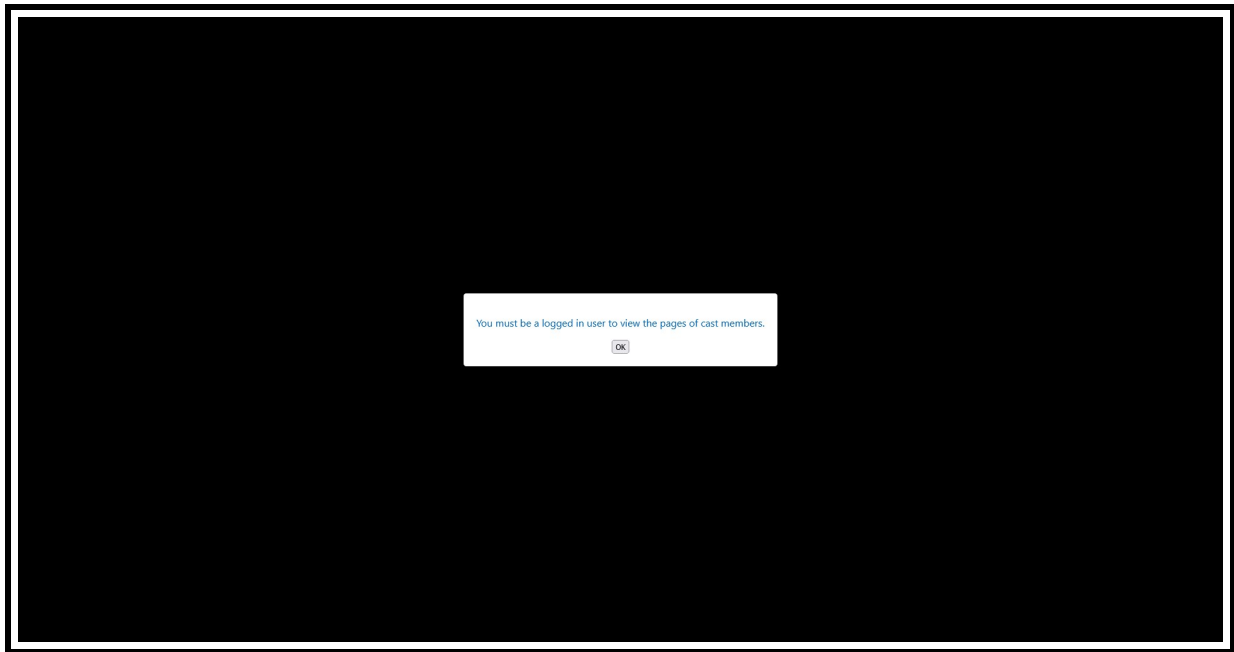
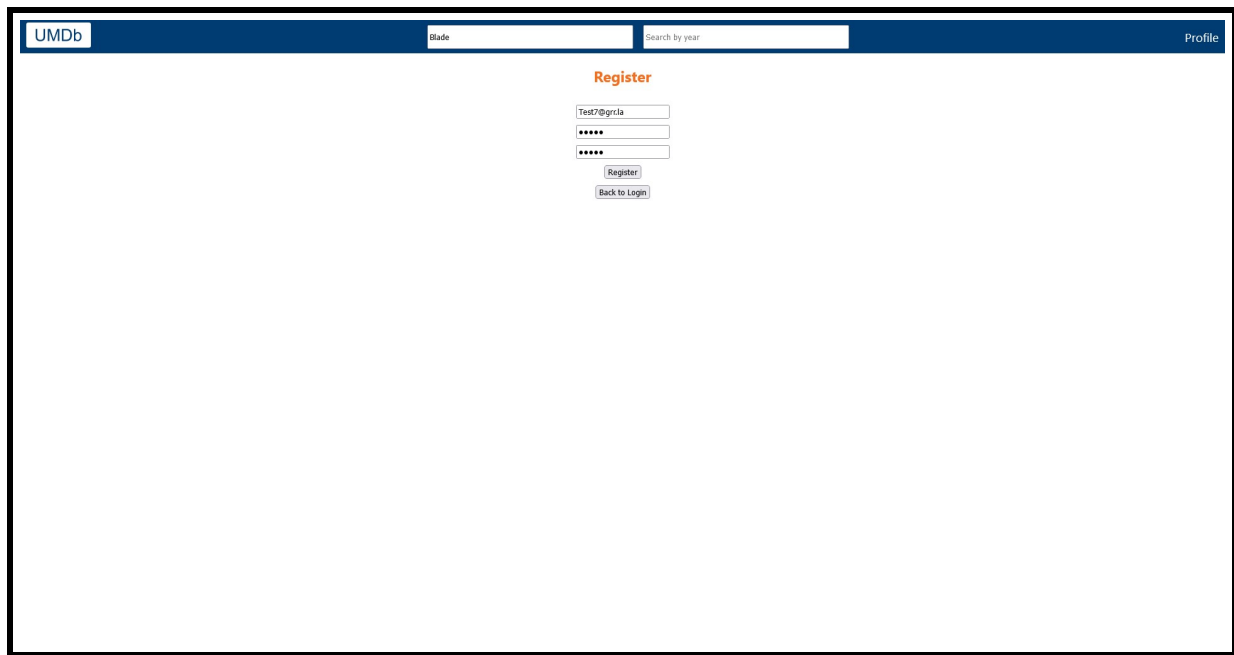


Figure 17: Test 5 - Click on a Cast Member (Logged Out)

UMDb	Blade	Search by year:	Profile
Person Details			
Ray Liotta			
Date of Birth: 1954			
Death: 2022			
Roles:			
Movie Name	Character	IMDb Rating	
Goodfellas	Henry Hill	8.7	
Unlawful Entry	Officer Pete Davis	6.4	
Corrina, Corrina	Manny Singer	6.6	
No Escape	Capt. J.T. Robbins	6.1	
Operation Dumbo Drop	Capt. T.C. Doyle	5.2	
Unforgettable	David Krane	6	
Cop Land	Gary Figgis	7	
Turbulence	Ryan Weaver	4.9	

Figure 18: Test 6 - Click on a Cast Member (Logged In)



UMDb

Blade

Search by year

Profile

**Register**

Test7@gr1a

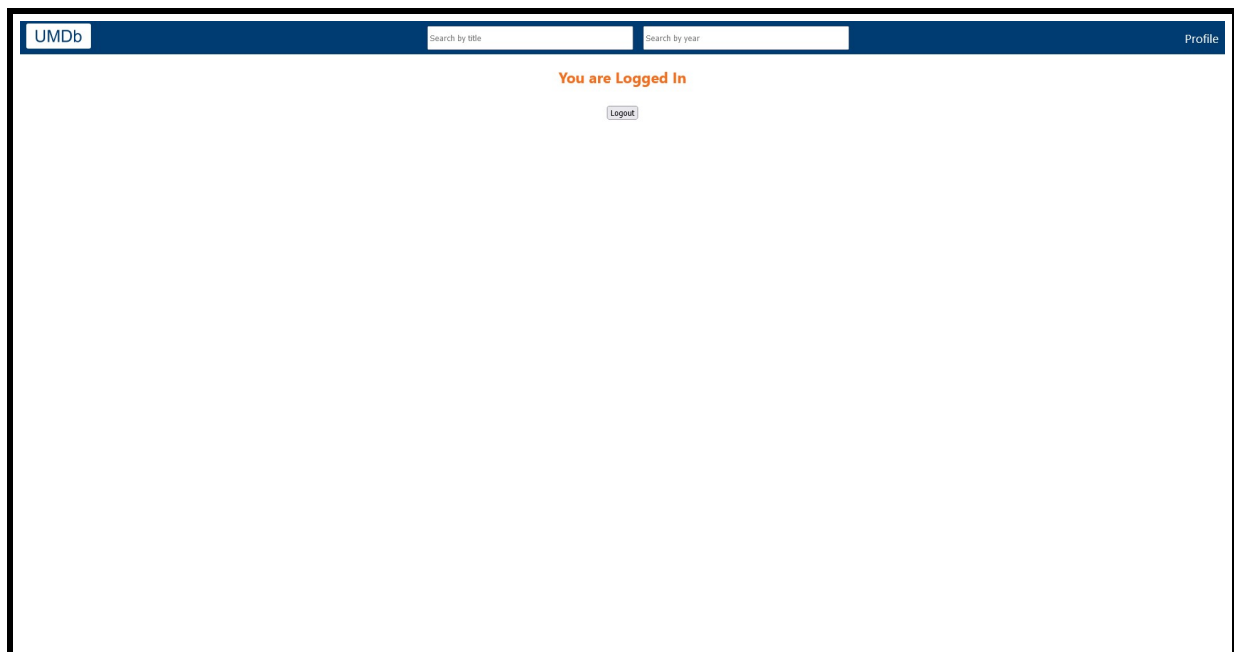
\*\*\*\*\*

\*\*\*\*\*

Register

Back to Login

Figure 19: Test 7 - Register a New User



UMDb

Search by title

Search by year

Profile

**You are Logged In**

Logout

Figure 20: Test 7 - Logged in as a User

The screenshot shows the UMDb login page. At the top, there is a dark blue header bar with the UMDb logo on the left, a search bar in the center, and a 'Profile' link on the right. Below the header, the word 'Login' is displayed in orange. Underneath, there are two input fields: the first contains the email 'Test7@gr1a' and the second contains masked characters '\*\*\*\*\*'. Below these fields are two buttons: 'Login' and 'Register Now'.

Figure 21: Test 8 - Login in as User

The screenshot shows the UMDb page after a successful login. The header bar is identical to the previous figure, but the search bar now has 'Search by title' and 'Search by year' options. In the center of the page, the text 'You are Logged In' is displayed in orange. Below this text is a 'Logout' button.

Figure 22: Test 8 - Logged in as a User

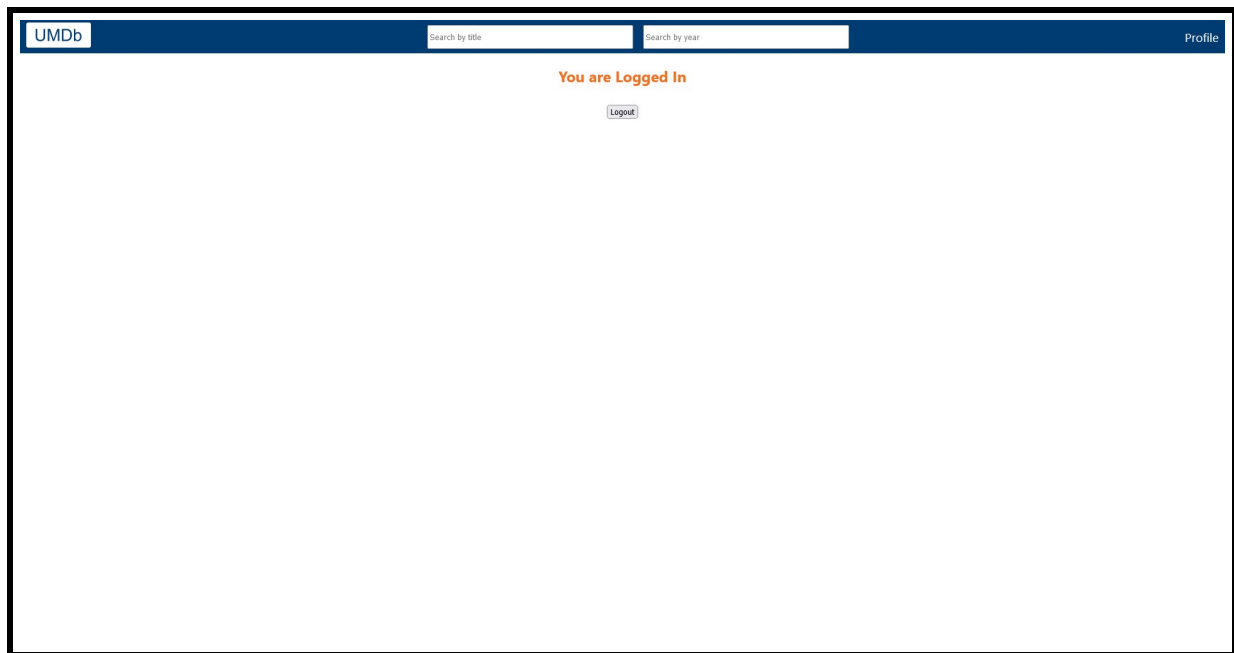


Figure 23: Test 9 - Ready to Logout

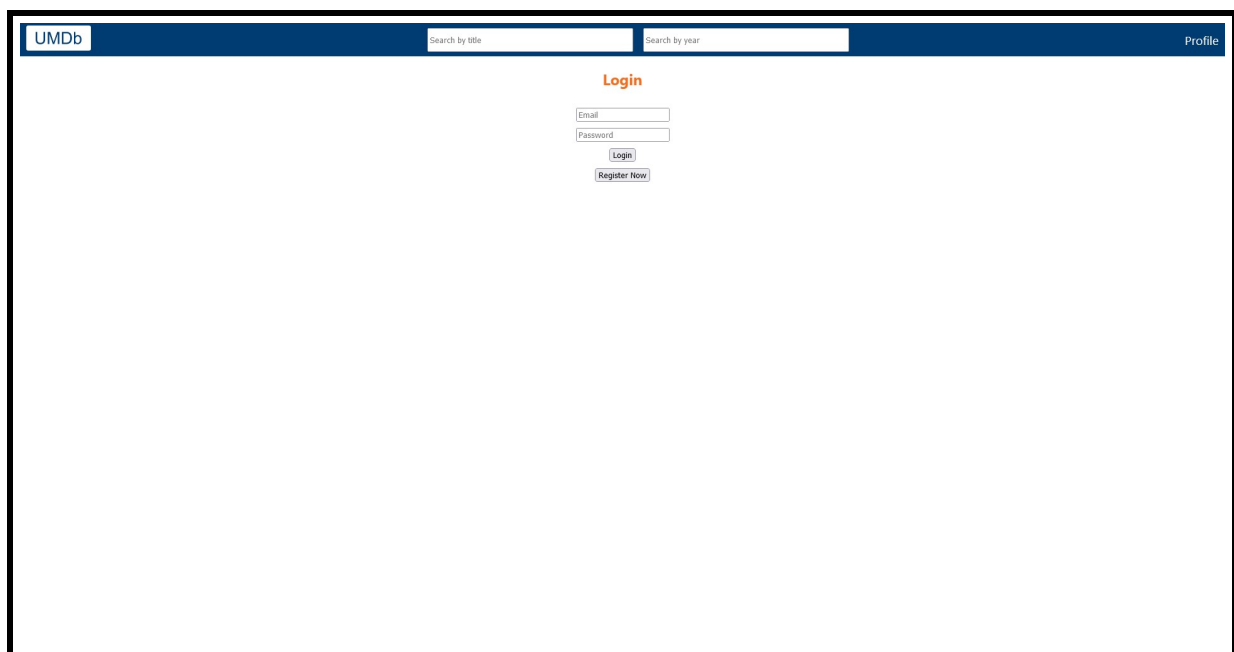


Figure 24: Test 9 - Logged Out